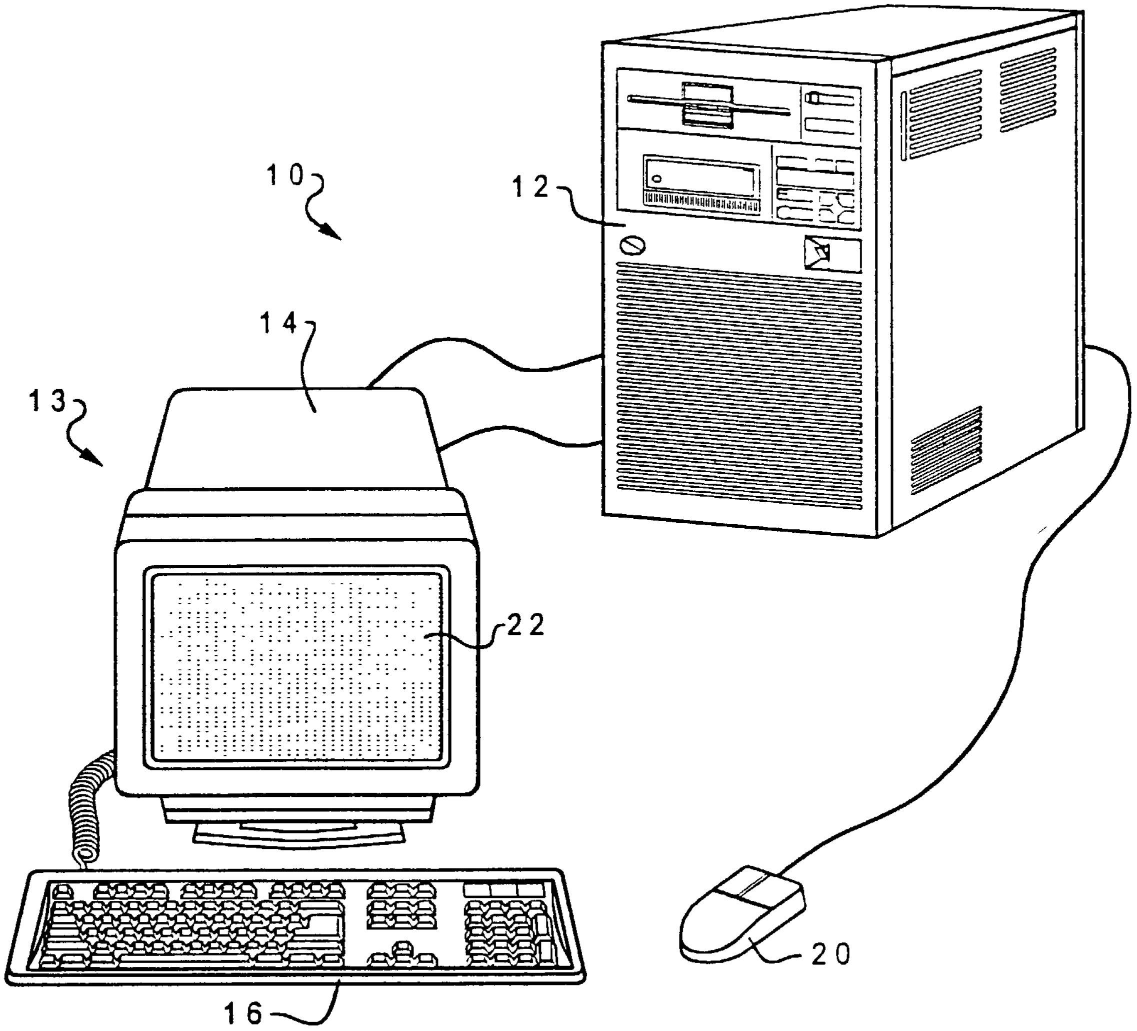


Fig. 1

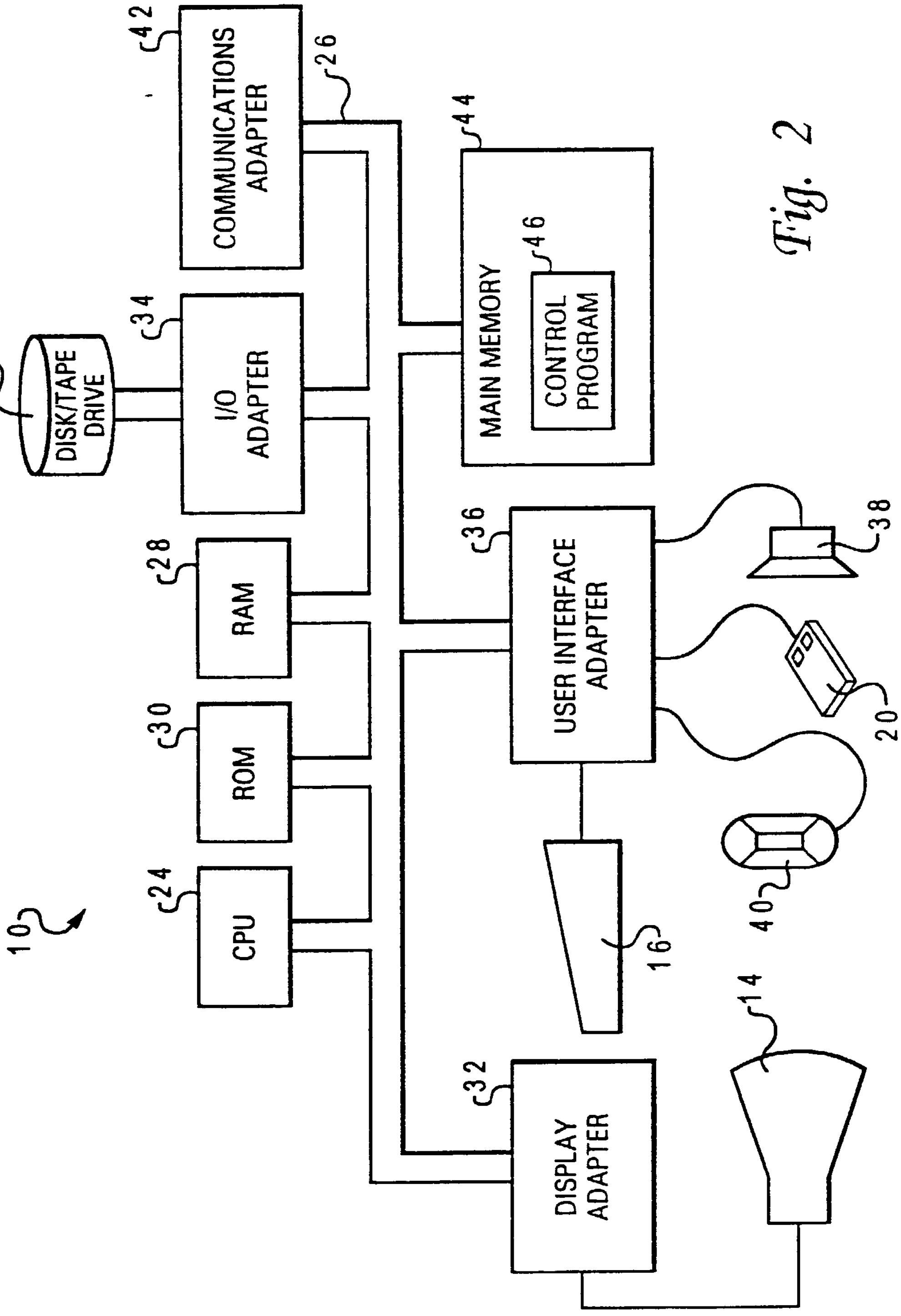

10
CPU
24
ROM
30
RAM
28
I/O ADAPTER
34
DISK/TAPE DRIVE
33
COMMUNICATIONS ADAPTER
42
26
MAIN MEMORY
44
CONTROL PROGRAM
46
USER INTERFACE ADAPTER
36
38
20
40
16
DISPLAY ADAPTER
32
14

*Fig. 2*

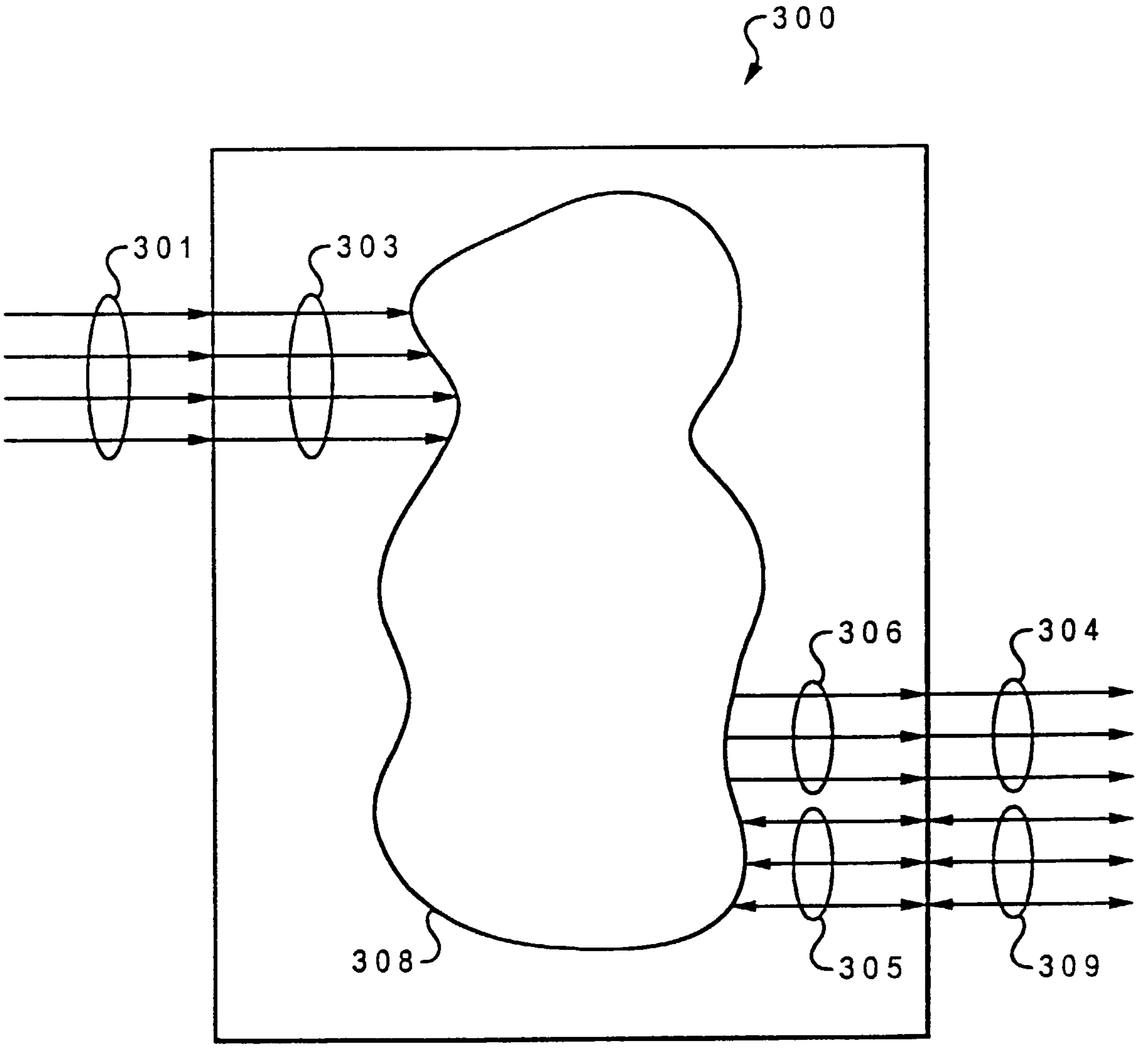


*Fig. 3A*

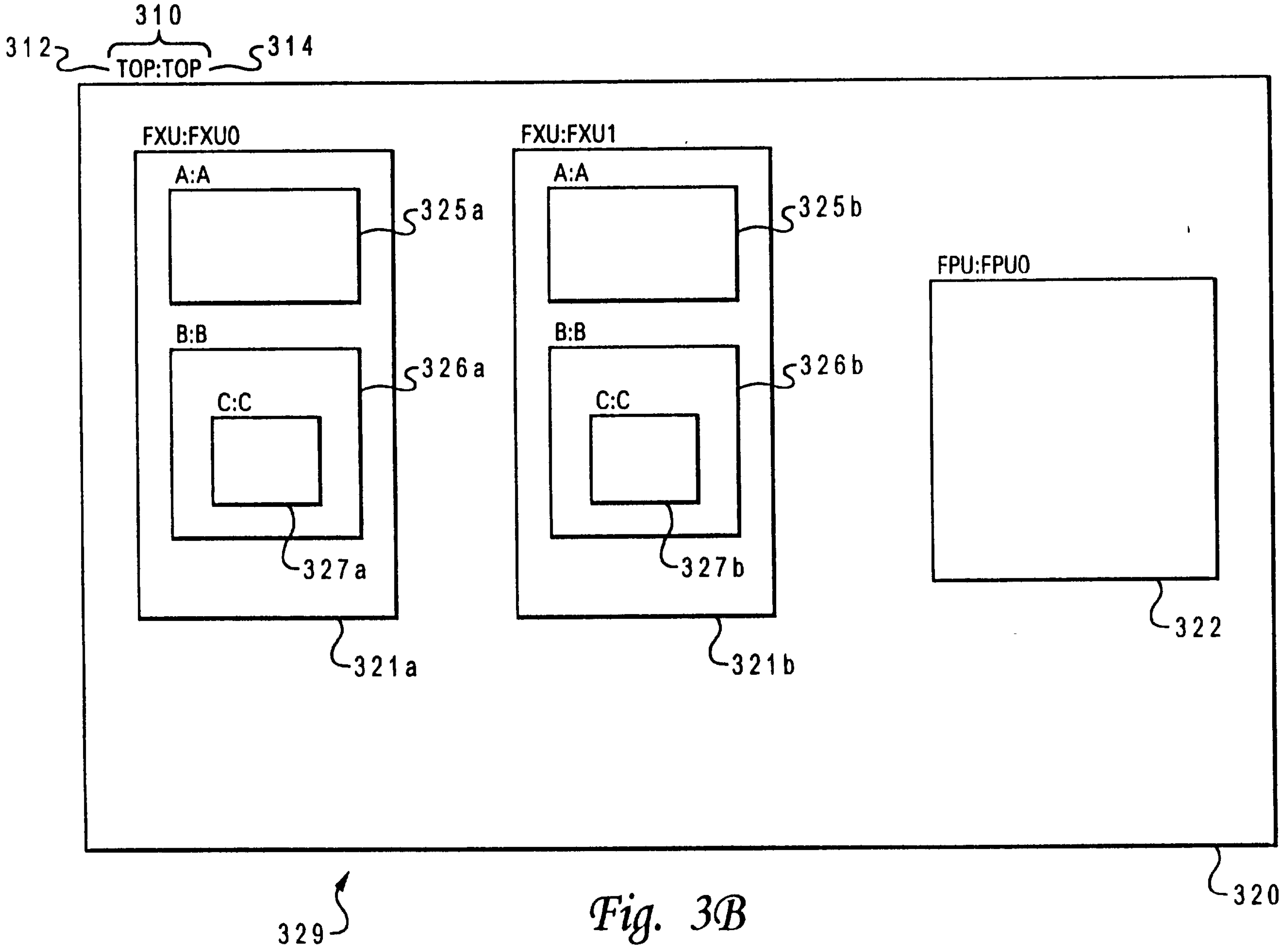

310
312
TOP:TOP
314
FXU:FXU0
A:A
325a
B:B
326a
C:C
327a
321a
FXU:FXU1
A:A
325b
B:B
326b
C:C
327b
321b
FPU:FPU0
322
320
329

*Fig. 3B*

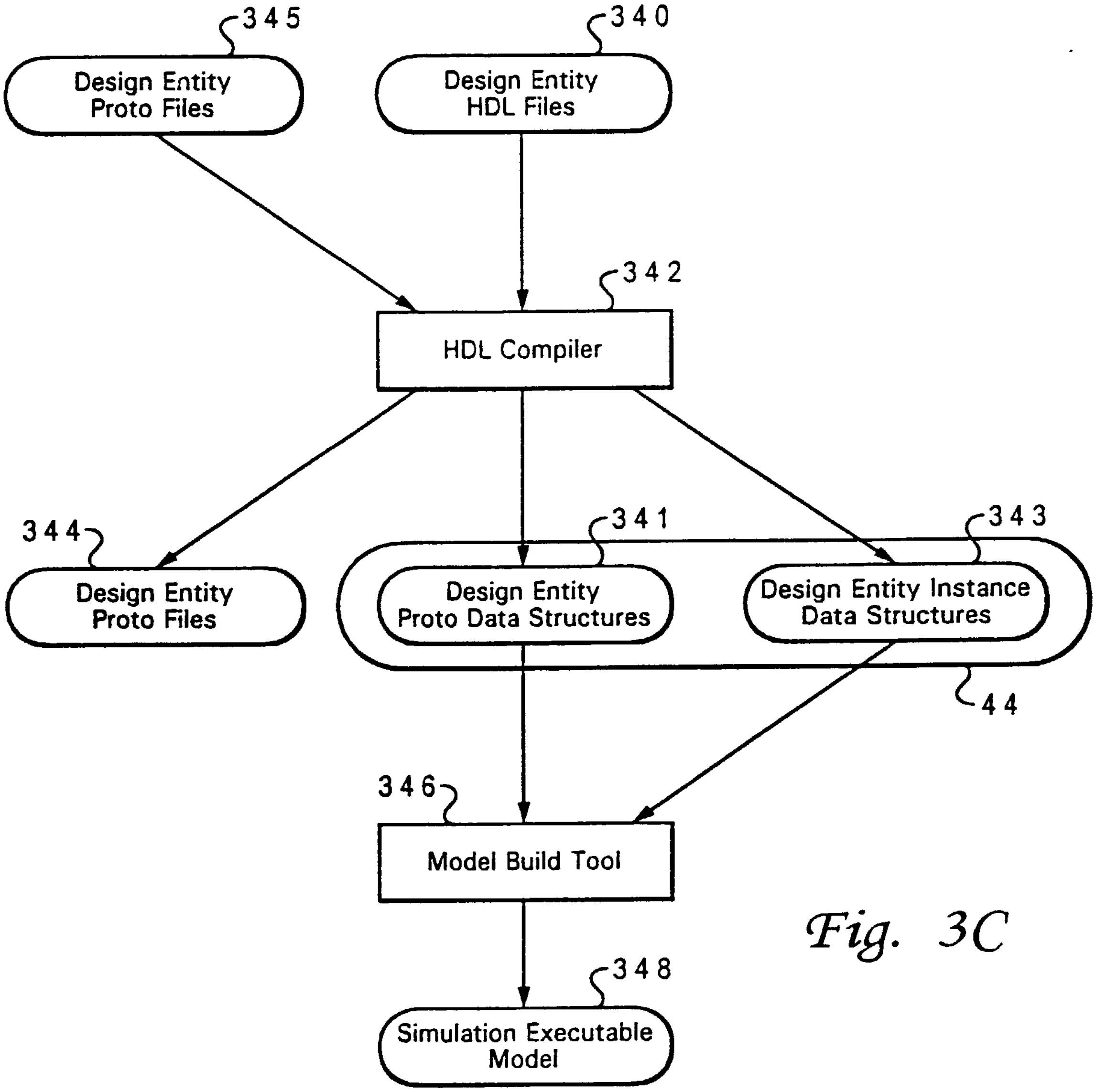

345
Design Entity Proto Files
340
Design Entity HDL Files
342
HDL Compiler
344
Design Entity Proto Files
341
Design Entity Proto Data Structures
343
Design Entity Instance Data Structures
44
346
Model Build Tool
348
Simulation Executable Model

*Fig. 3C*

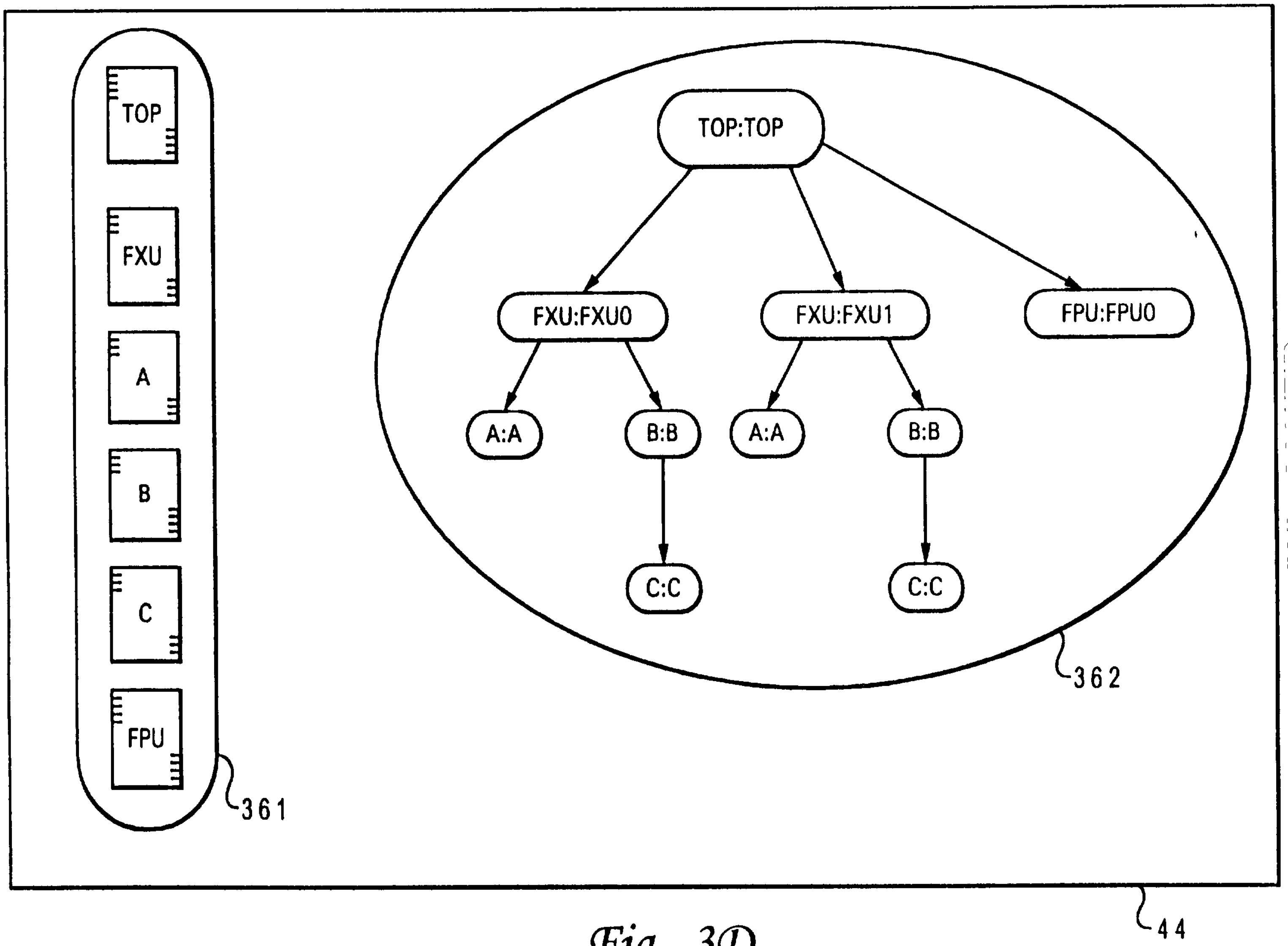

TOP
FXU
A
B
C
FPU
361
TOP:TOP
FXU:FXU0
FXU:FXU1
FPU:FPU0
A:A
B:B
A:A
B:B
C:C
C:C
362
44

Fig. 3D

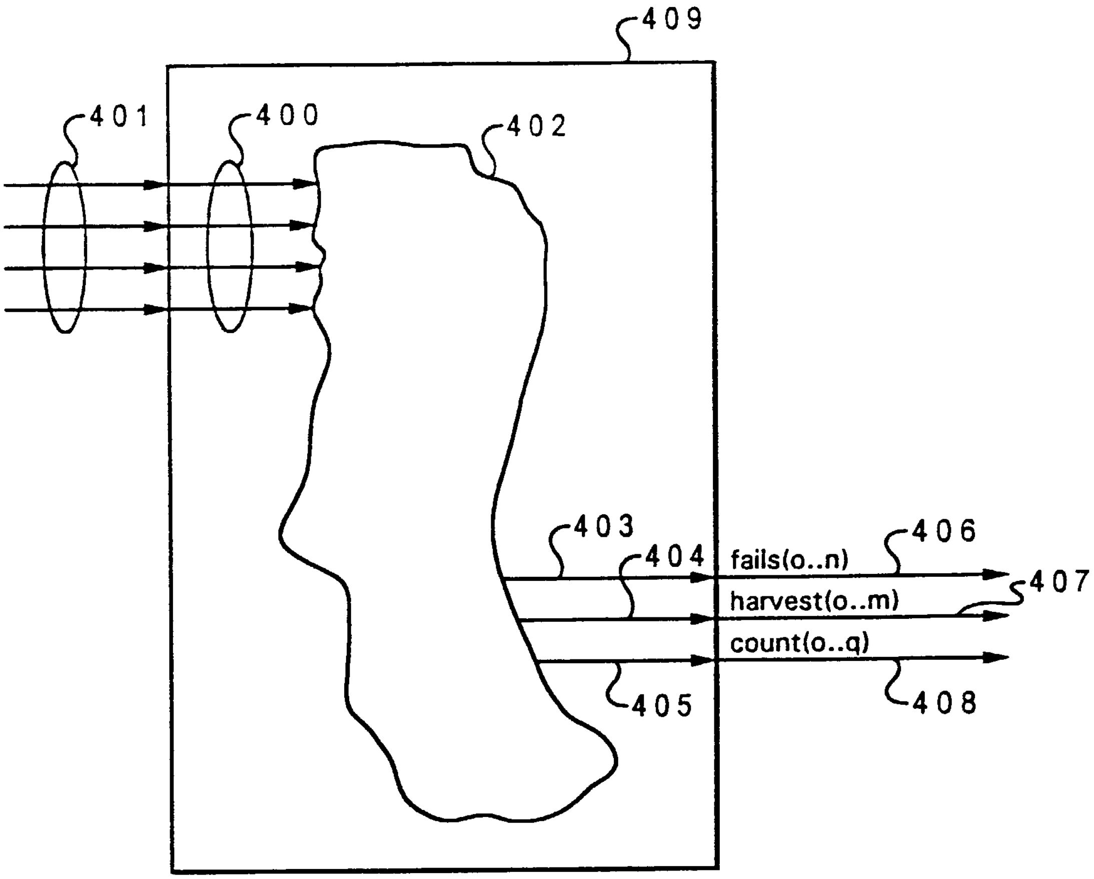

409
401
400
402
403
404
fails(o..n)
406
407
harvest(o..m)
count(o..q)
405
408

*Fig. 4A*

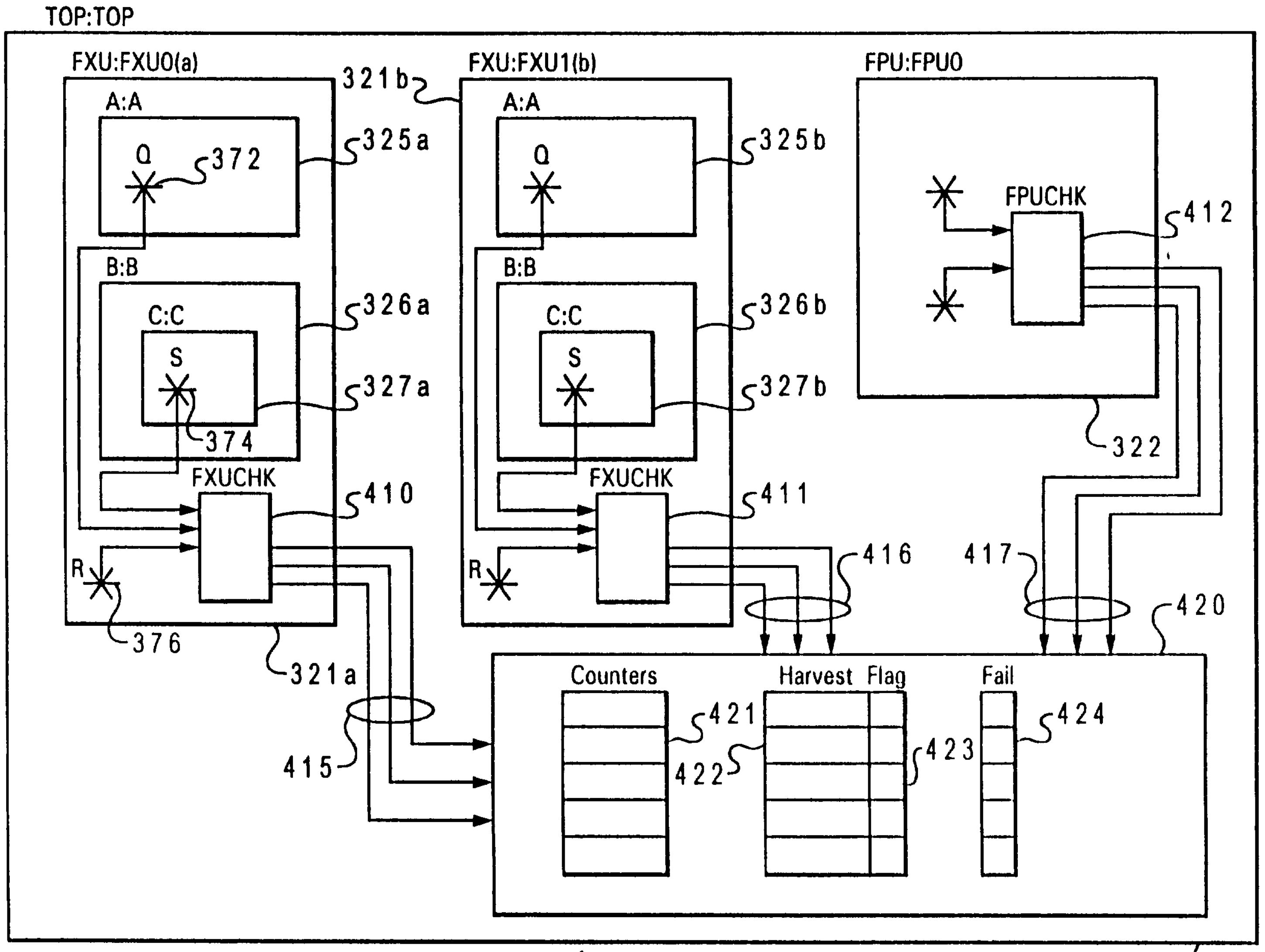
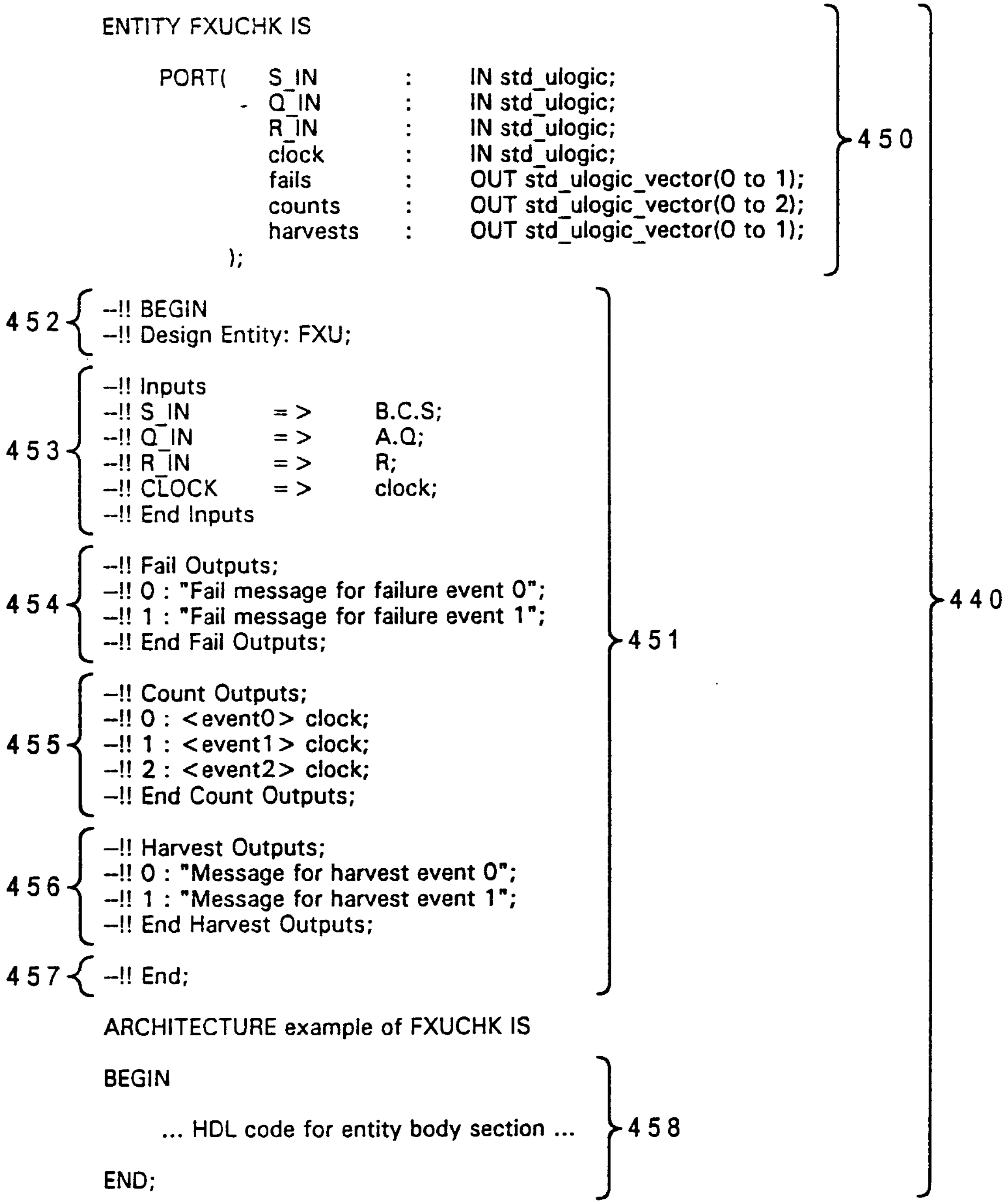

TOP:TOP
FXU:FXU0(a)
321b
FXU:FXU1(b)
FPU:FPU0
A:A
Q
372
325a
B:B
C:C
S
374
326a
327a
FXUCHK
410
R
376
321a
415
A:A
Q
325b
B:B
C:C
S
326b
327b
FXUCHK
411
R
416
FPUCHK
412
322
417
420
Counters
Harvest
Flag
Fail
421
422
423
424
329

*Fig. 4B*



```
ENTITY FXUCHK IS

     PORT(   S_IN        :     IN std_ulogic;
             Q_IN        :     IN std_ulogic;
             R_IN        :     IN std_ulogic;
             clock       :     IN std_ulogic;
             fails       :     OUT std_ulogic_vector(0 to 1);
             counts      :     OUT std_ulogic_vector(0 to 2);
             harvests    :     OUT std_ulogic_vector(0 to 1);
           );

--!! BEGIN
--!! Design Entity: FXU;

--!! Inputs
--!! S_IN        =>      B.C.S;
--!! Q_IN        =>      A.Q;
--!! R_IN        =>      R;
--!! CLOCK       =>      clock;
--!! End Inputs

--!! Fail Outputs;
--!! 0 : "Fail message for failure event 0";
--!! 1 : "Fail message for failure event 1";
--!! End Fail Outputs;

--!! Count Outputs;
--!! 0 : <event0> clock;
--!! 1 : <event1> clock;
--!! 2 : <event2> clock;
--!! End Count Outputs;

--!! Harvest Outputs;
--!! 0 : "Message for harvest event 0";
--!! 1 : "Message for harvest event 1";
--!! End Harvest Outputs;

--!! End;

ARCHITECTURE example of FXUCHK IS

BEGIN

     ... HDL code for entity body section ...

END;
```

*Fig. 4C*

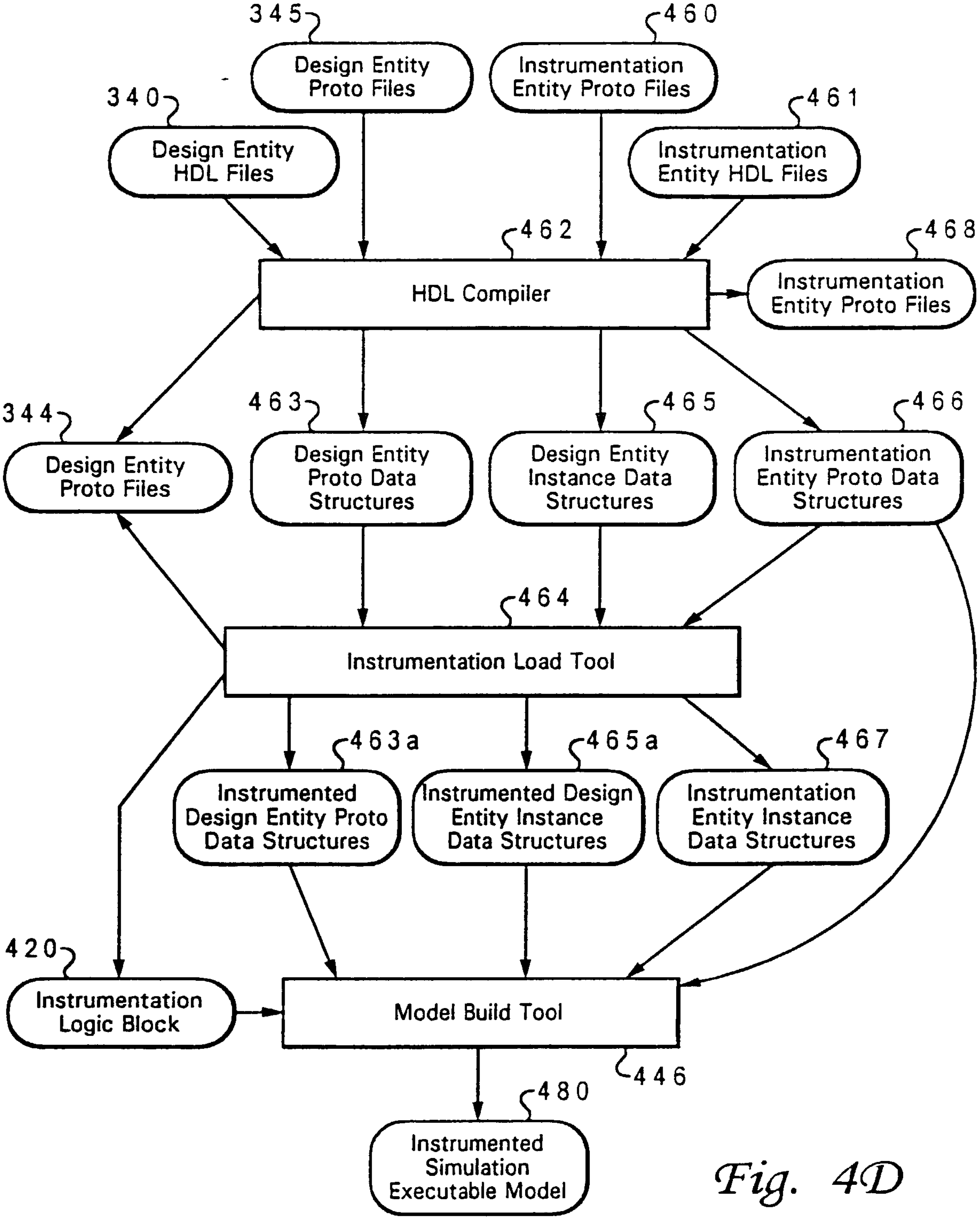

345
Design Entity Proto Files
460
Instrumentation Entity Proto Files
340
Design Entity HDL Files
461
Instrumentation Entity HDL Files
462
HDL Compiler
468
Instrumentation Entity Proto Files
344
Design Entity Proto Files
463
Design Entity Proto Data Structures
465
Design Entity Instance Data Structures
466
Instrumentation Entity Proto Data Structures
464
Instrumentation Load Tool
463a
Instrumented Design Entity Proto Data Structures
465a
Instrumented Design Entity Instance Data Structures
467
Instrumentation Entity Instance Data Structures
420
Instrumentation Logic Block
Model Build Tool
446
480
Instrumented Simulation Executable Model

*Fig. 4D*

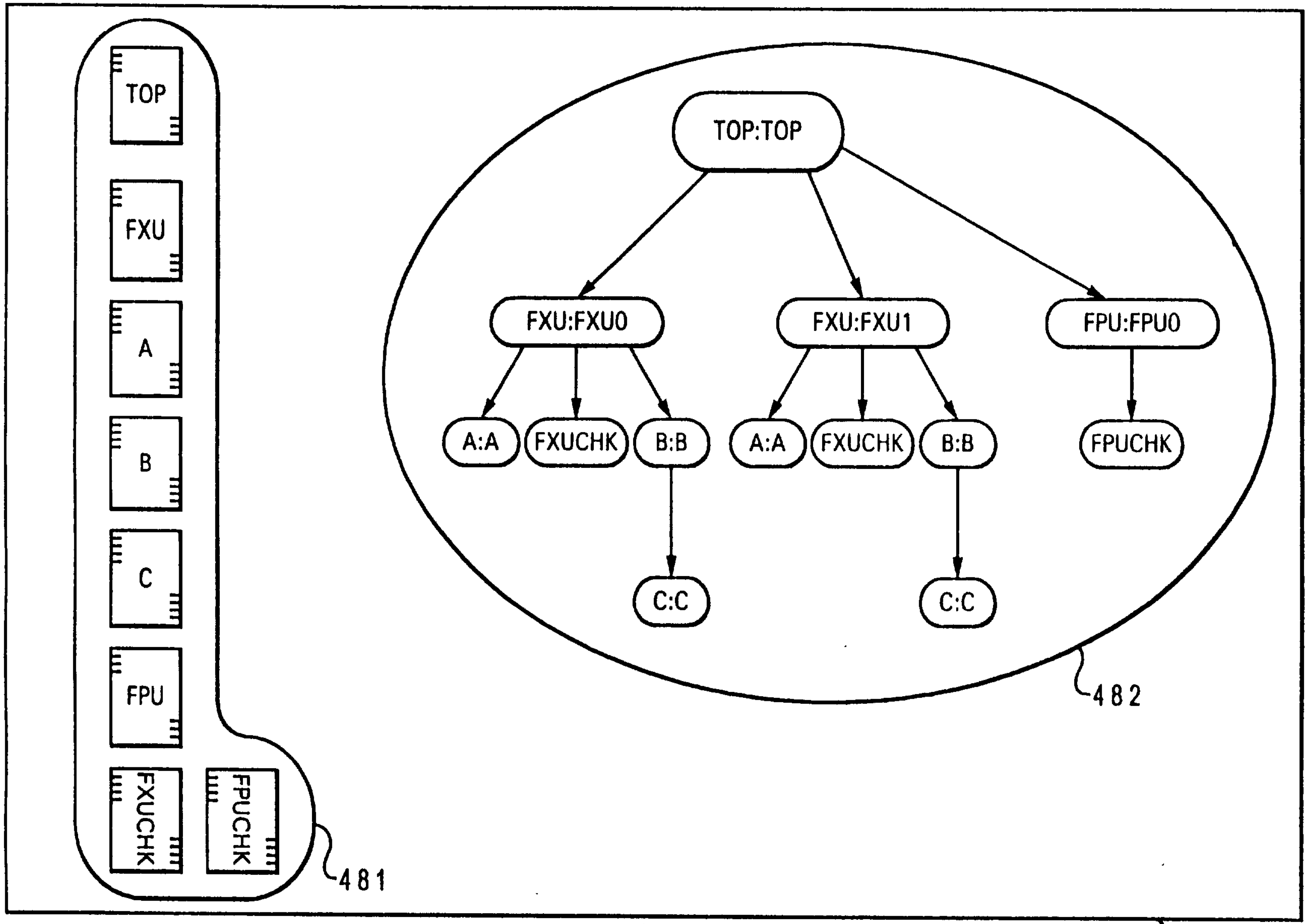

TOP
FXU
A
B
C
FPU
FXUCHK
FPUCHK
481
TOP:TOP
FXU:FXU0
FXU:FXU1
FPU:FPU0
A:A
FXUCHK
B:B
A:A
FXUCHK
B:B
FPUCHK
C:C
C:C
482
44

*Fig. 4E*

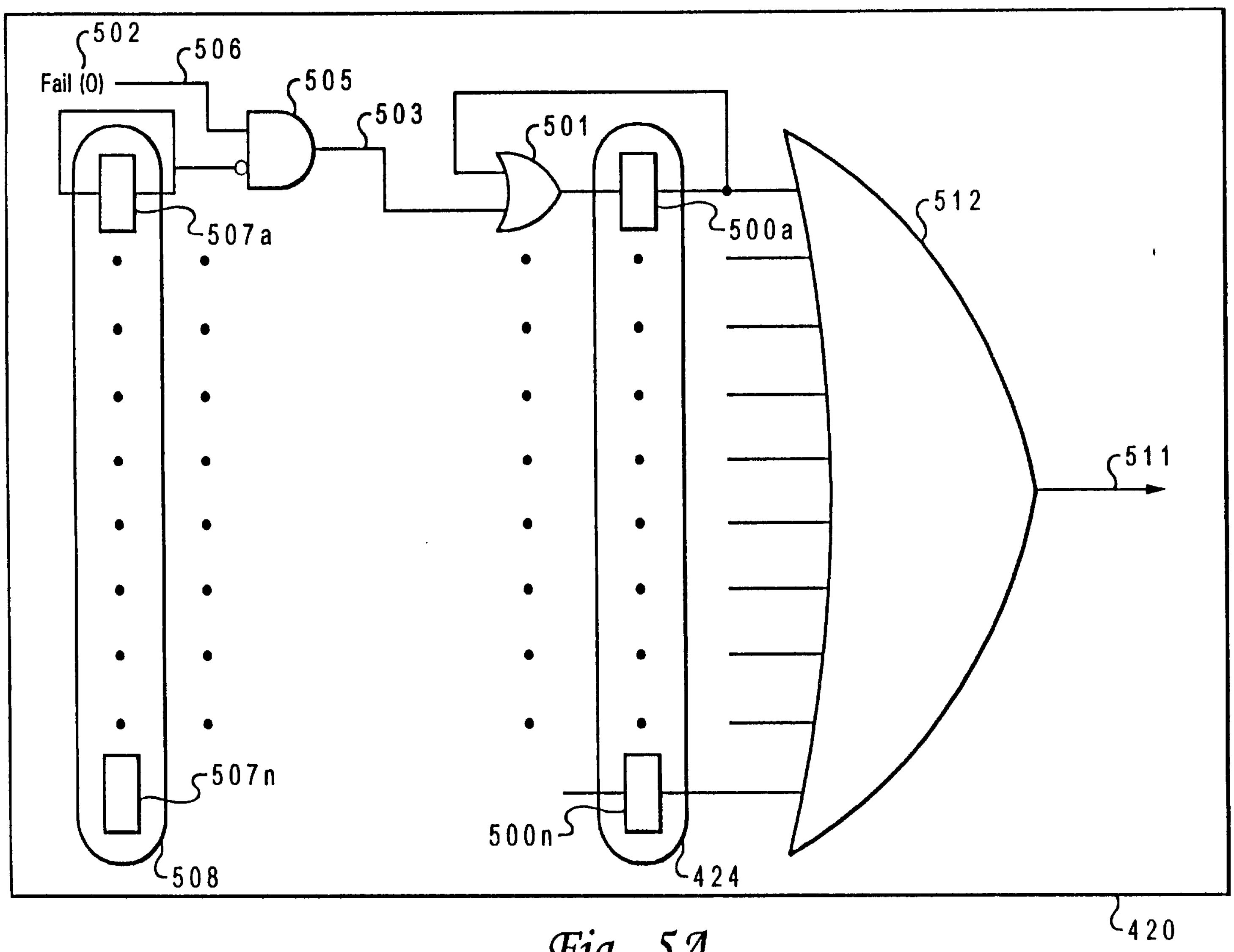

502
Fail (0)
506
505
503
501
507a
500a
512
511
507n
500n
508
424
420

Fig. 5A

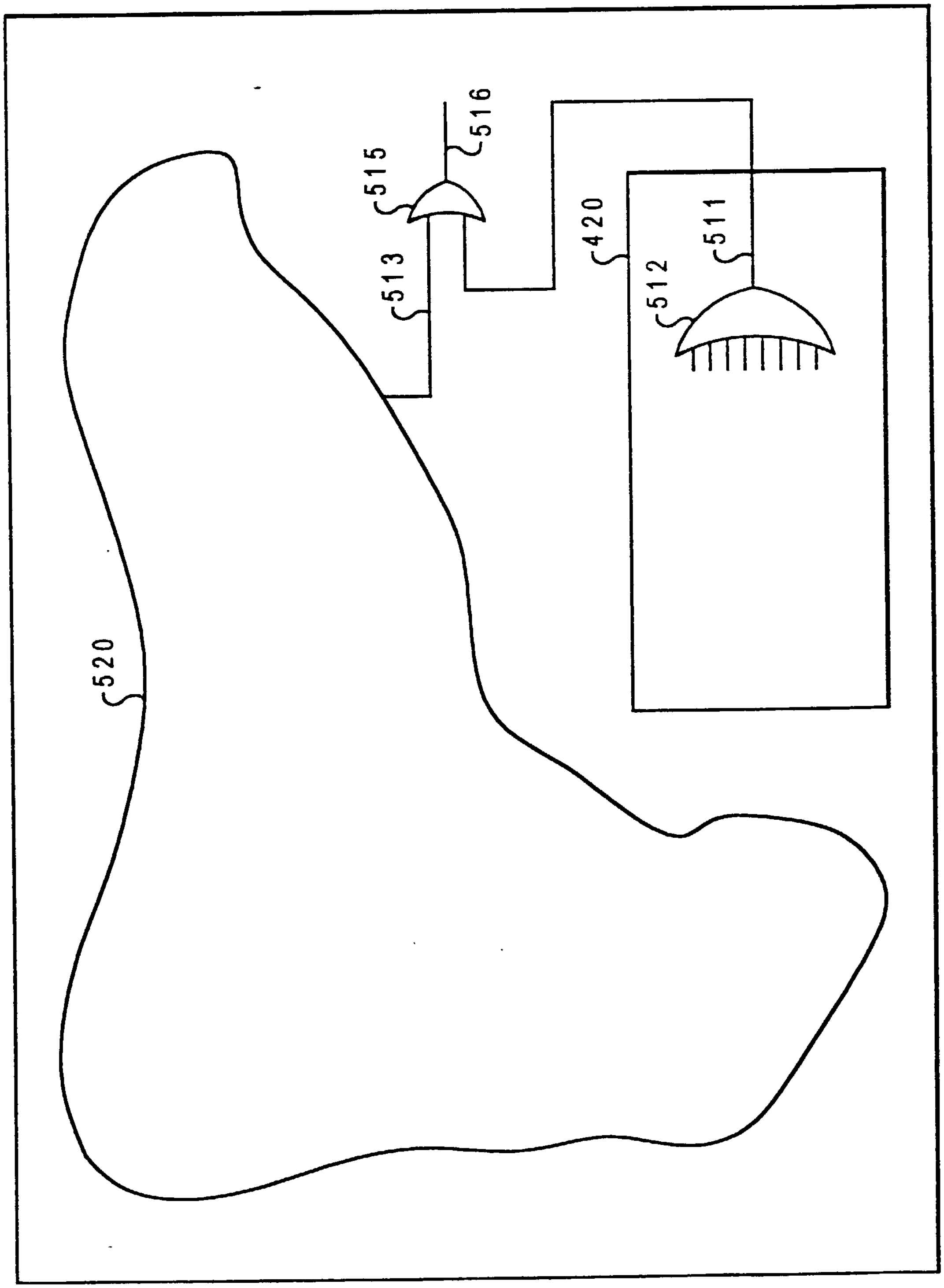


*Fig. 5B*

[redacted]

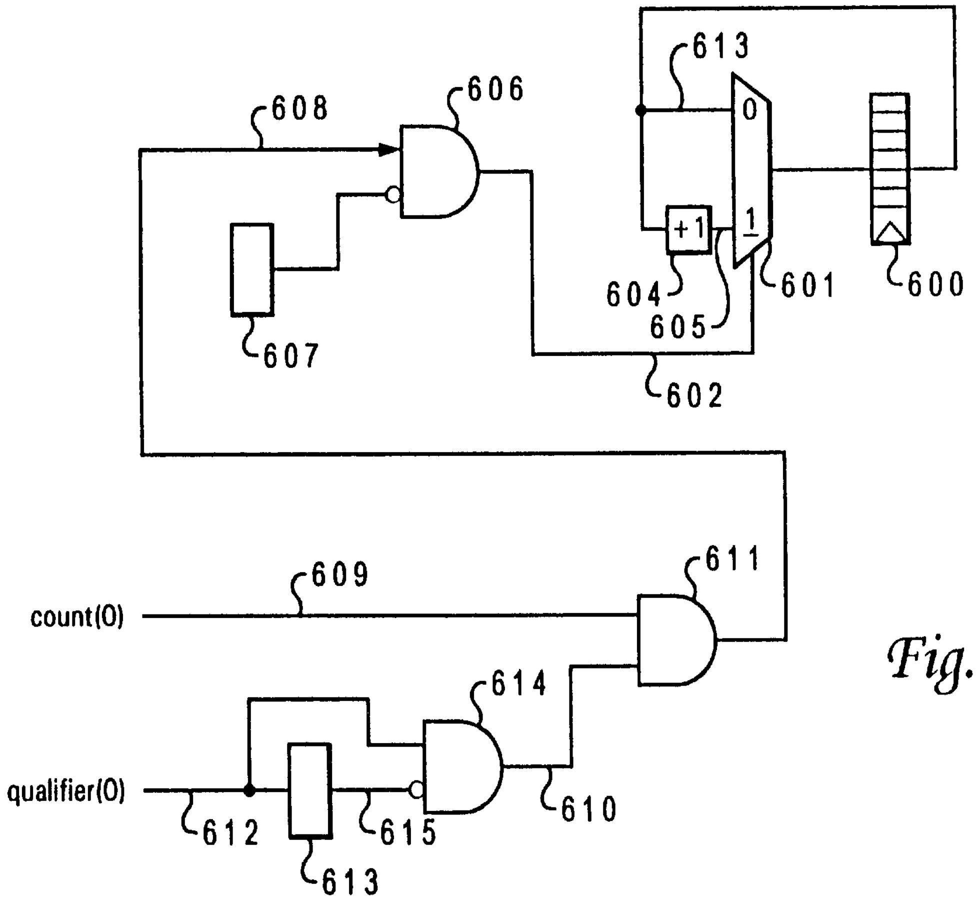


Fig. 6A

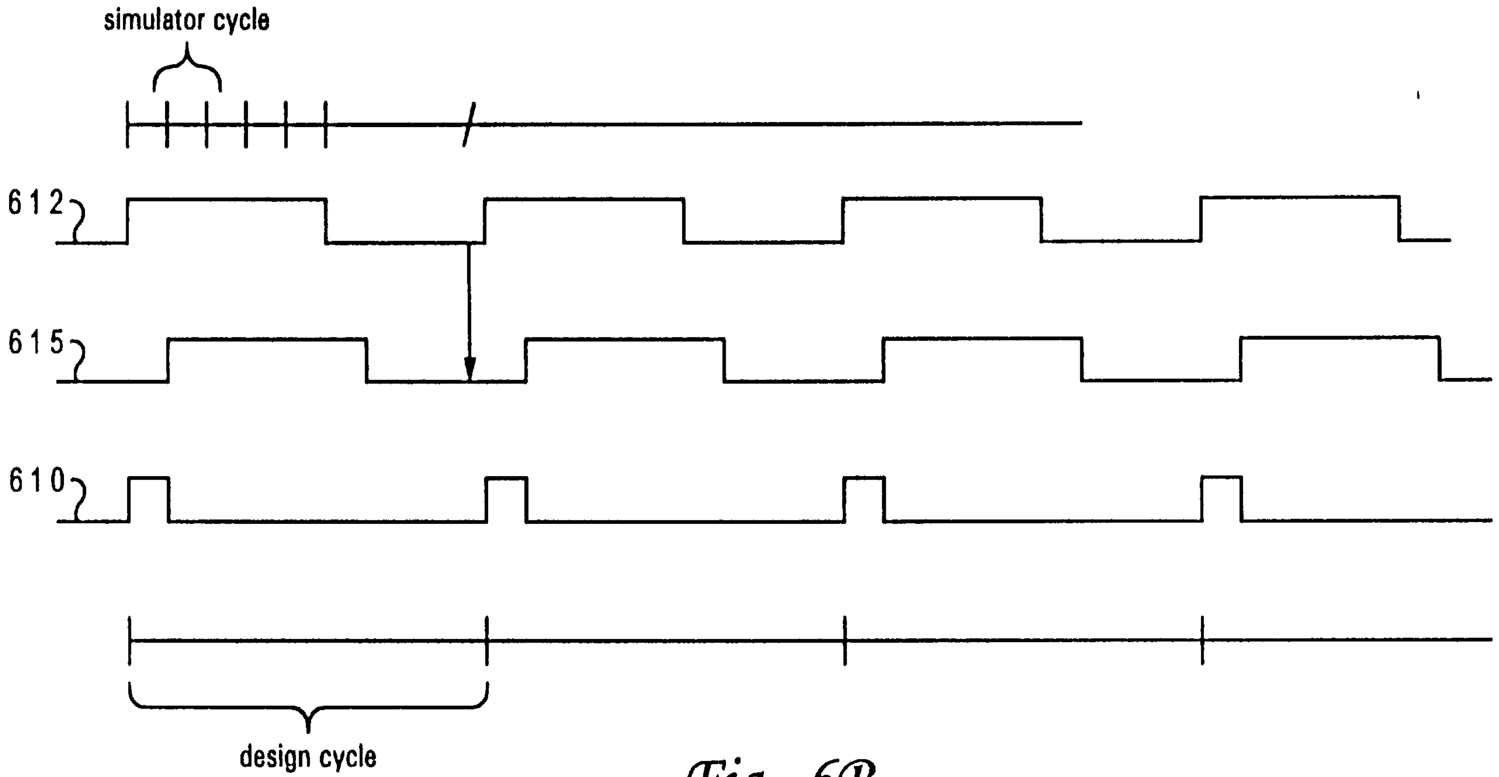


Fig. 6B

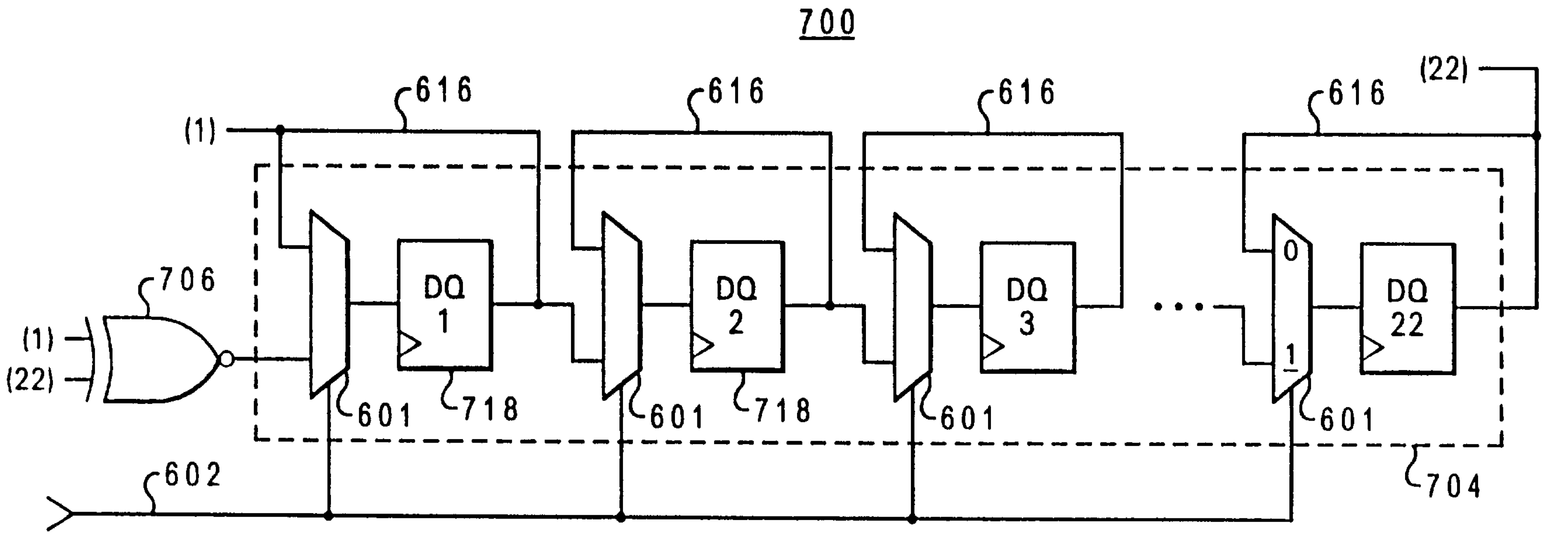

700
(1)
(22)
616
616
616
616
(22)
706
(1)
(22)
DQ
1
DQ
2
DQ
3
DQ
22
0
1
601
718
601
718
601
601
704
602

Fig. 7

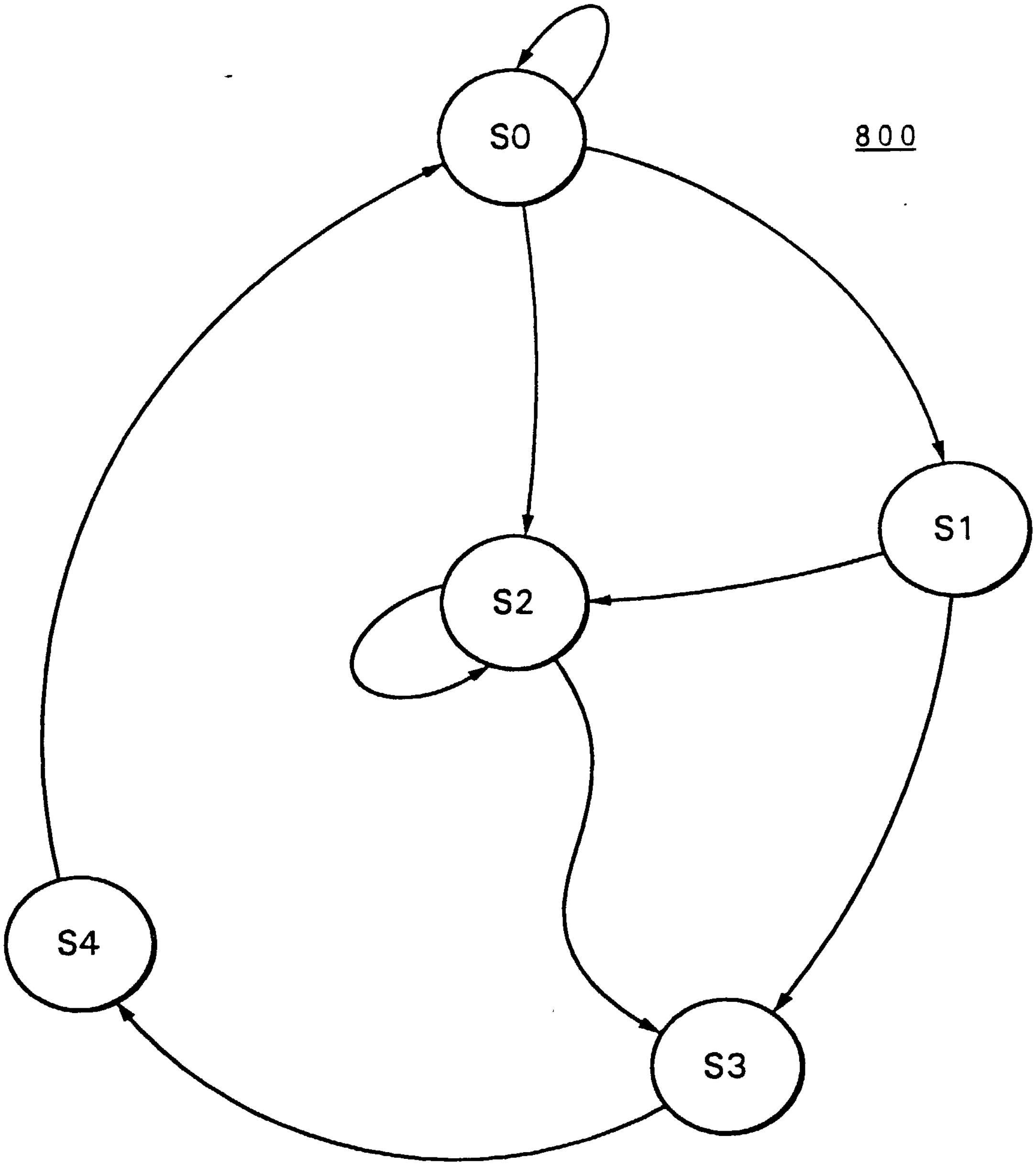


*Fig. 8A*
*Prior Art*

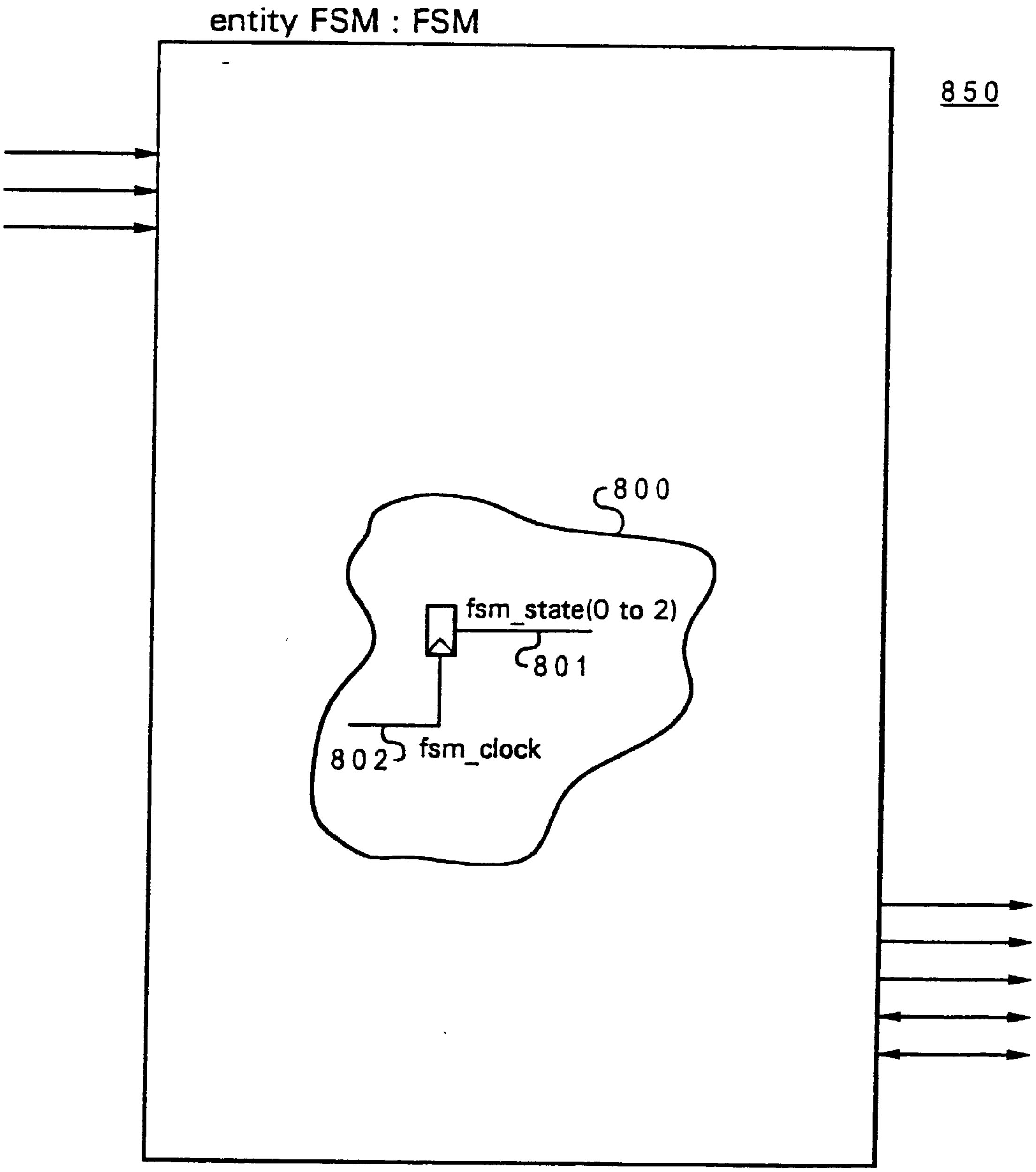
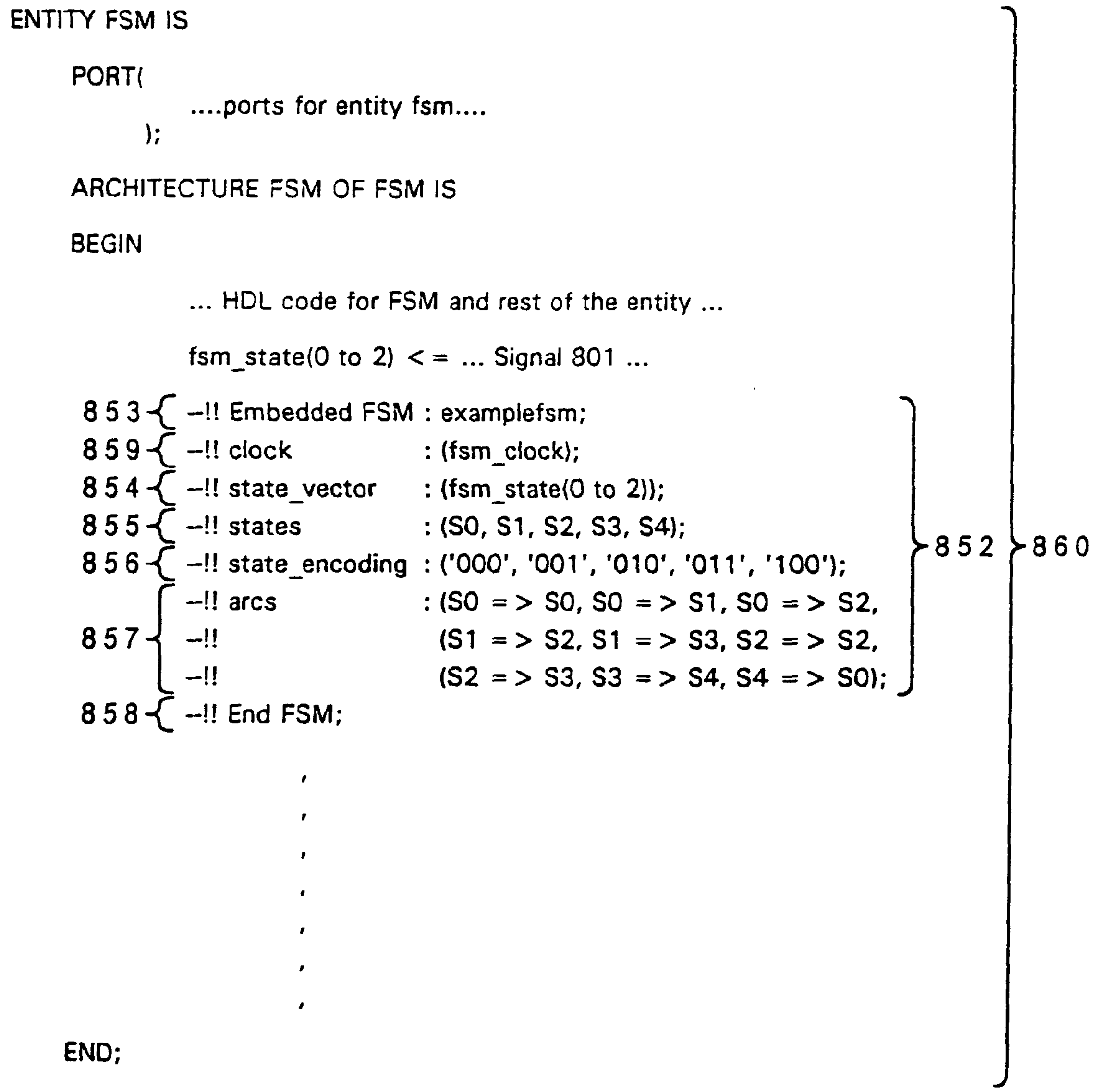


Fig. 8B
Prior Art



```
ENTITY FSM IS

    PORT(
                ....ports for entity fsm....
          );

    ARCHITECTURE FSM OF FSM IS

    BEGIN

                ... HDL code for FSM and rest of the entity ...

                fsm_state(0 to 2) <= ... Signal 801 ...

          --!! Embedded FSM  : examplefsm;
          --!! clock         : (fsm_clock);
          --!! state_vector  : (fsm_state(0 to 2));
          --!! states        : (S0, S1, S2, S3, S4);
          --!! state_encoding : ('000', '001', '010', '011', '100');
          --!! arcs          : (S0 => S0, S0 => S1, S0 => S2,
          --!!                  (S1 => S2, S1 => S3, S2 => S2,
          --!!                  (S2 => S3, S3 => S4, S4 => S0);
          --!! End FSM;

                      ,
                      ,
                      ,
                      ,
                      ,
                      ,
                      ,

    END;
```

853
859
854
855
856
857
858
852
860

*Fig. 8C*

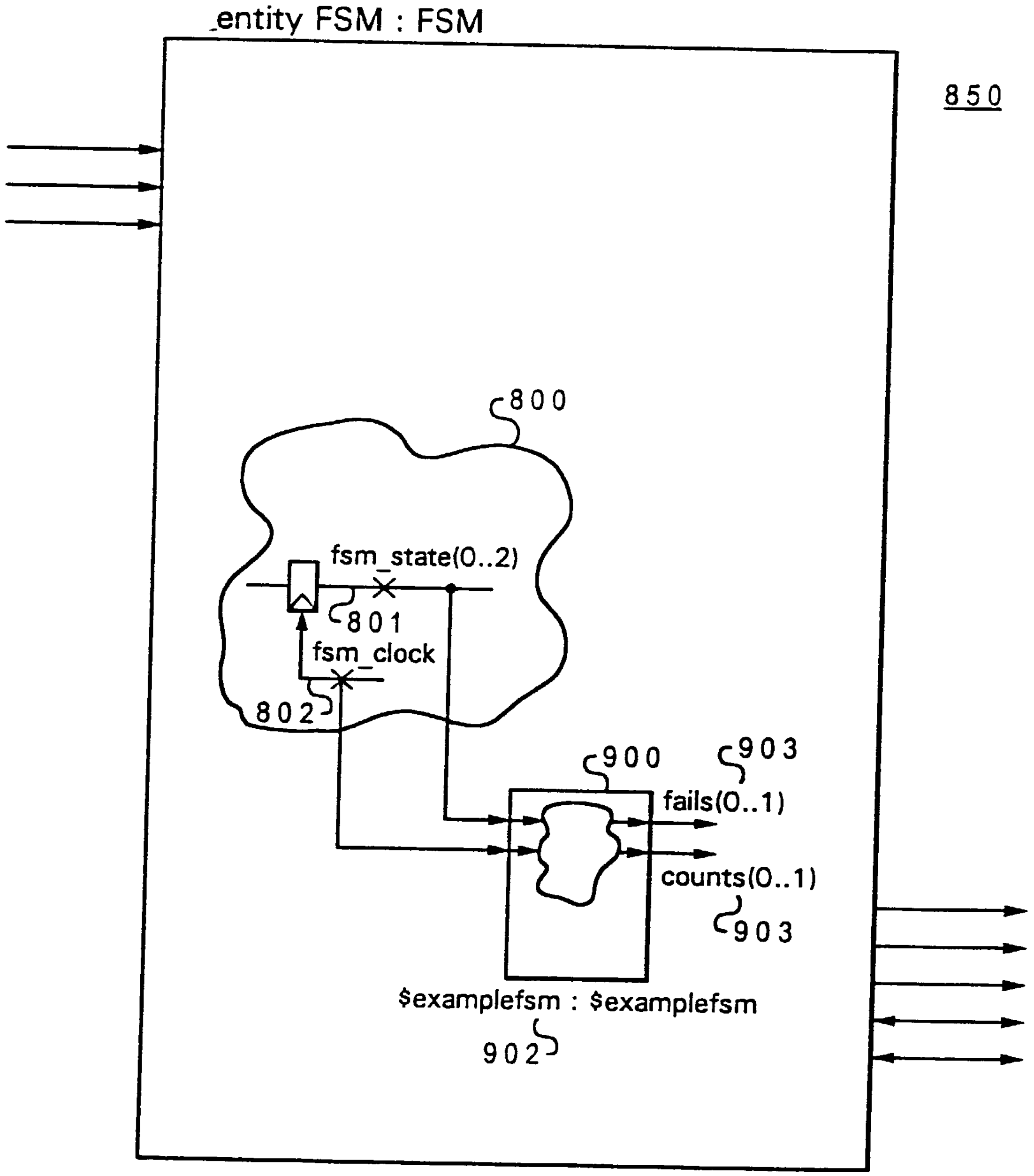

entity FSM : FSM
850
800
fsm_state(0..2)
801
fsm_clock
802
900
903
fails(0..1)
counts(0..1)
903
$examplefsm : $examplefsm
902

Fig. 9

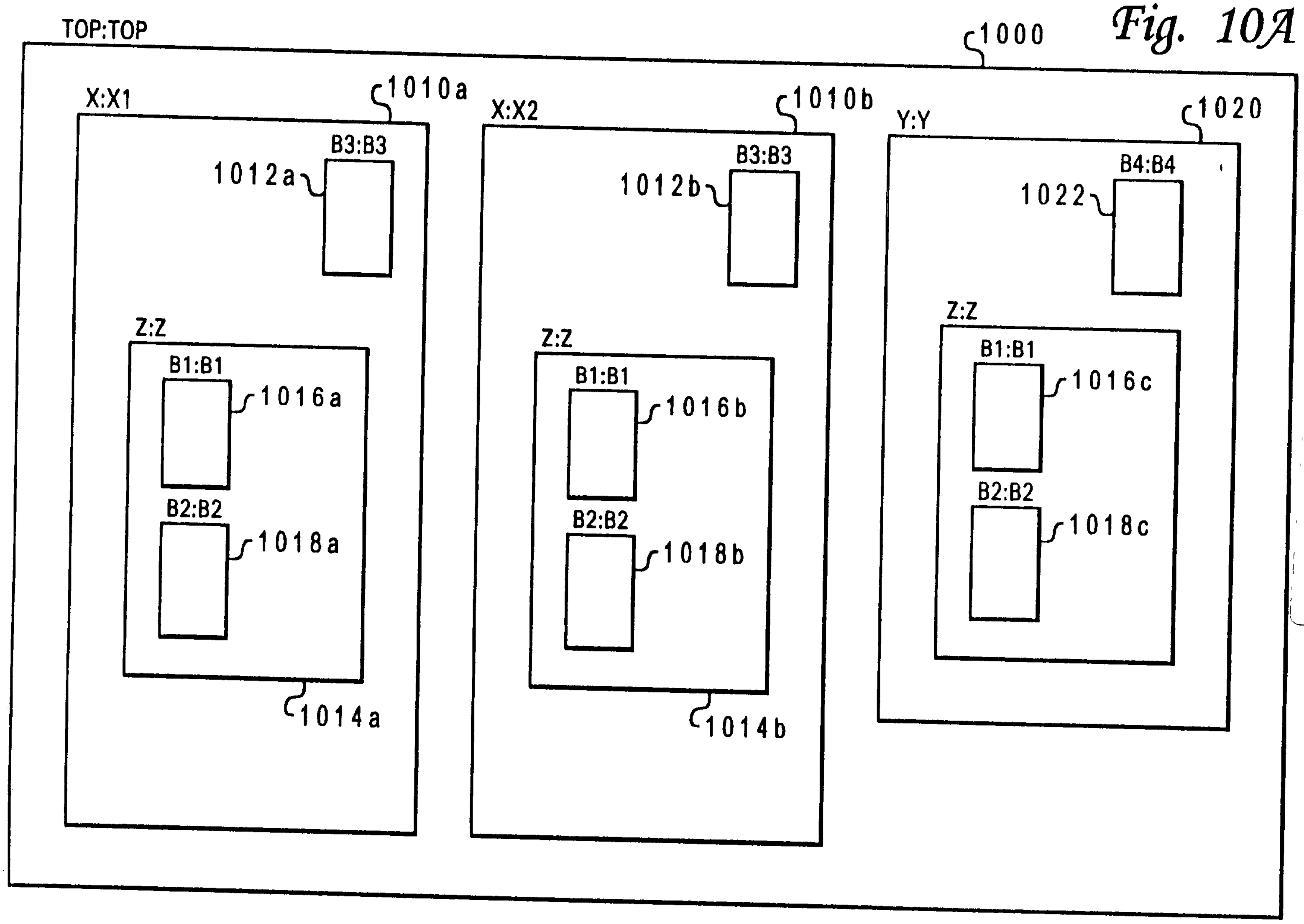
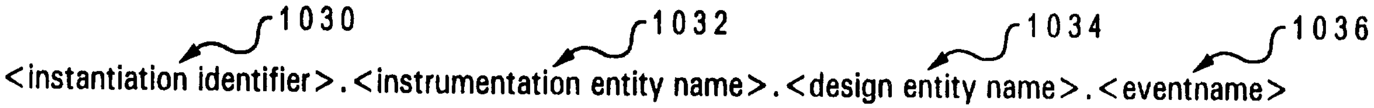
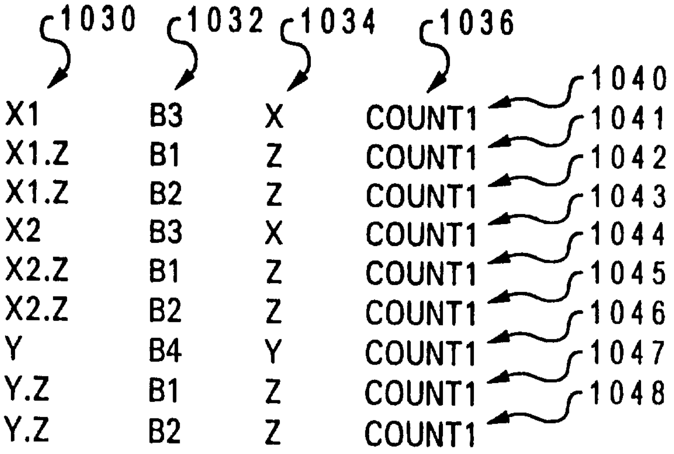

TOP:TOP
1000
X:X1
1010a
B3:B3
1012a
Z:Z
B1:B1
1016a
B2:B2
1018a
1014a
X:X2
1010b
B3:B3
1012b
Z:Z
B1:B1
1016b
B2:B2
1018b
1014b
Y:Y
1020
B4:B4
1022
Z:Z
B1:B1
1016c
B2:B2
1018c

*Fig. 10A*



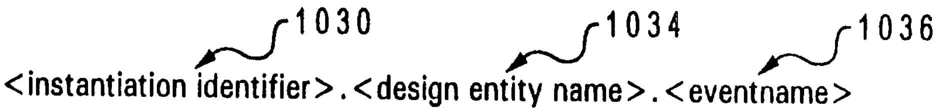
<instantiation identifier>.<instrumentation entity name>.<design entity name>.<eventname>

Fig. 10B



| | | | |
|---|---|---|---|
| X1 | B3 | X | COUNT1 |
| X1.Z | B1 | Z | COUNT1 |
| X1.Z | B2 | Z | COUNT1 |
| X2 | B3 | X | COUNT1 |
| X2.Z | B1 | Z | COUNT1 |
| X2.Z | B2 | Z | COUNT1 |
| Y | B4 | Y | COUNT1 |
| Y.Z | B1 | Z | COUNT1 |
| Y.Z | B2 | Z | COUNT1 |

Fig. 10C



<instantiation identifier>.<design entity name>.<eventname>

Fig. 10D

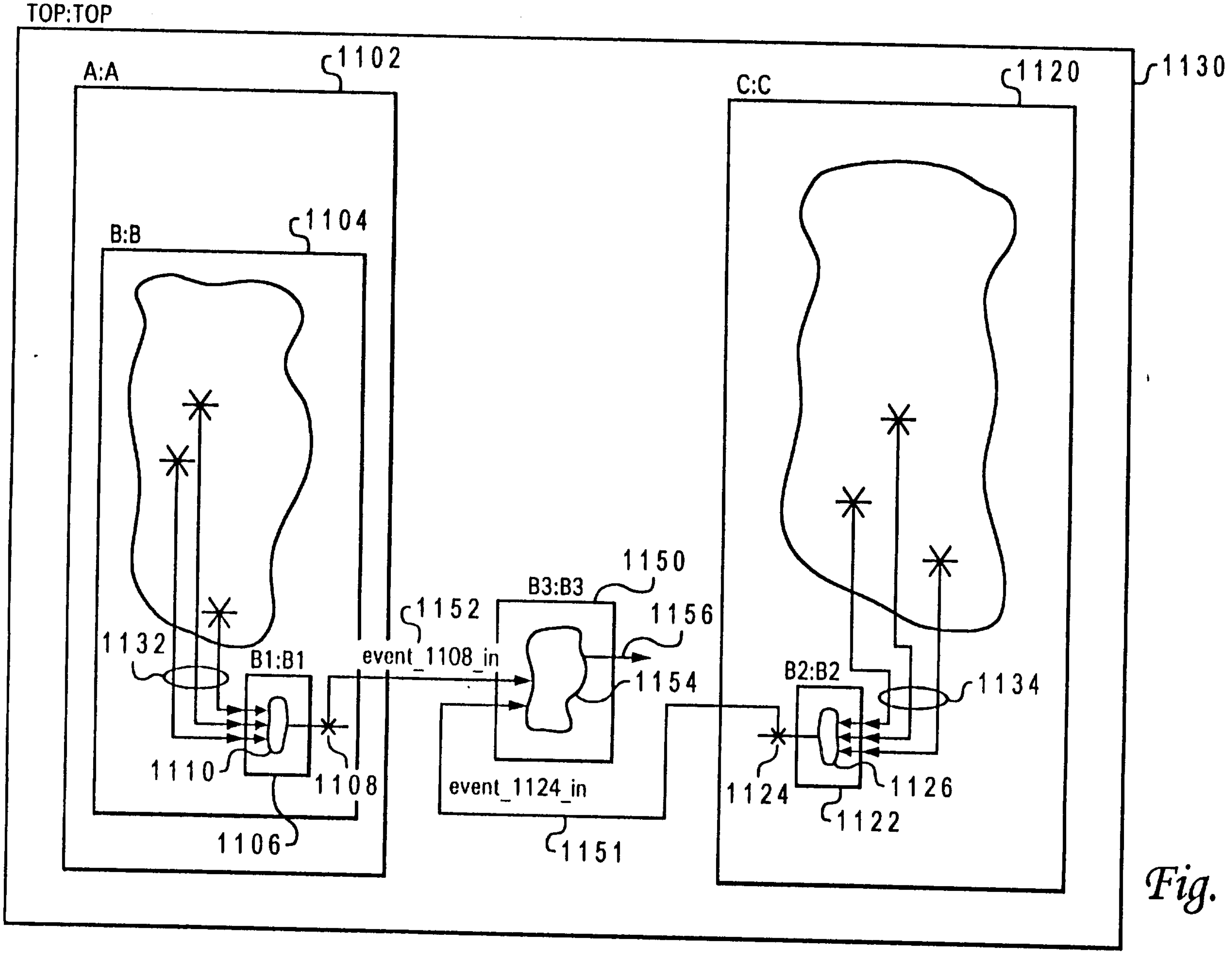
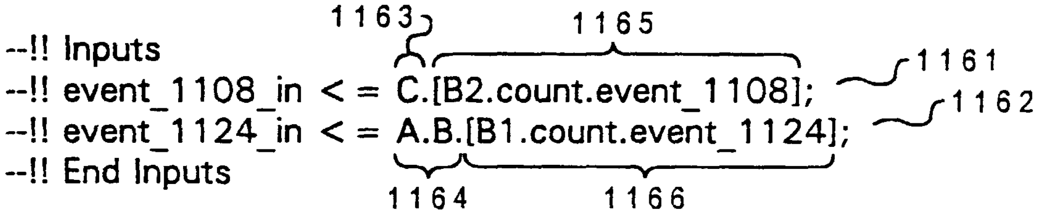
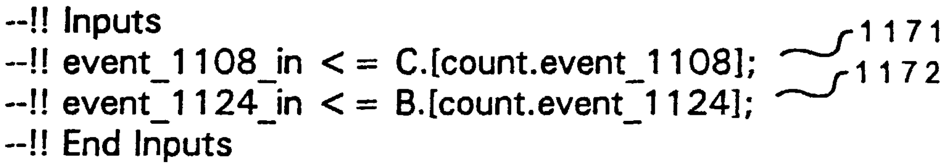

TOP:TOP
1130
A:A
1102
B:B
1104
C:C
1120
1132
B1:B1
1110
1108
1106
event_1108_in
1152
B3:B3
1150
1156
1154
event_1124_in
1151
B2:B2
1134
1124
1126
1122

*Fig. 11A*



```
--!! Inputs
--!! event_1108_in <= C.[B2.count.event_1108];
--!! event_1124_in <= A.B.[B1.count.event_1124];
--!! End Inputs
```

*Fig. 11B*



```
--!! Inputs
--!! event_1108_in <= C.[count.event_1108];
--!! event_1124_in <= B.[count.event_1124];
--!! End Inputs
```

*Fig. 11C*

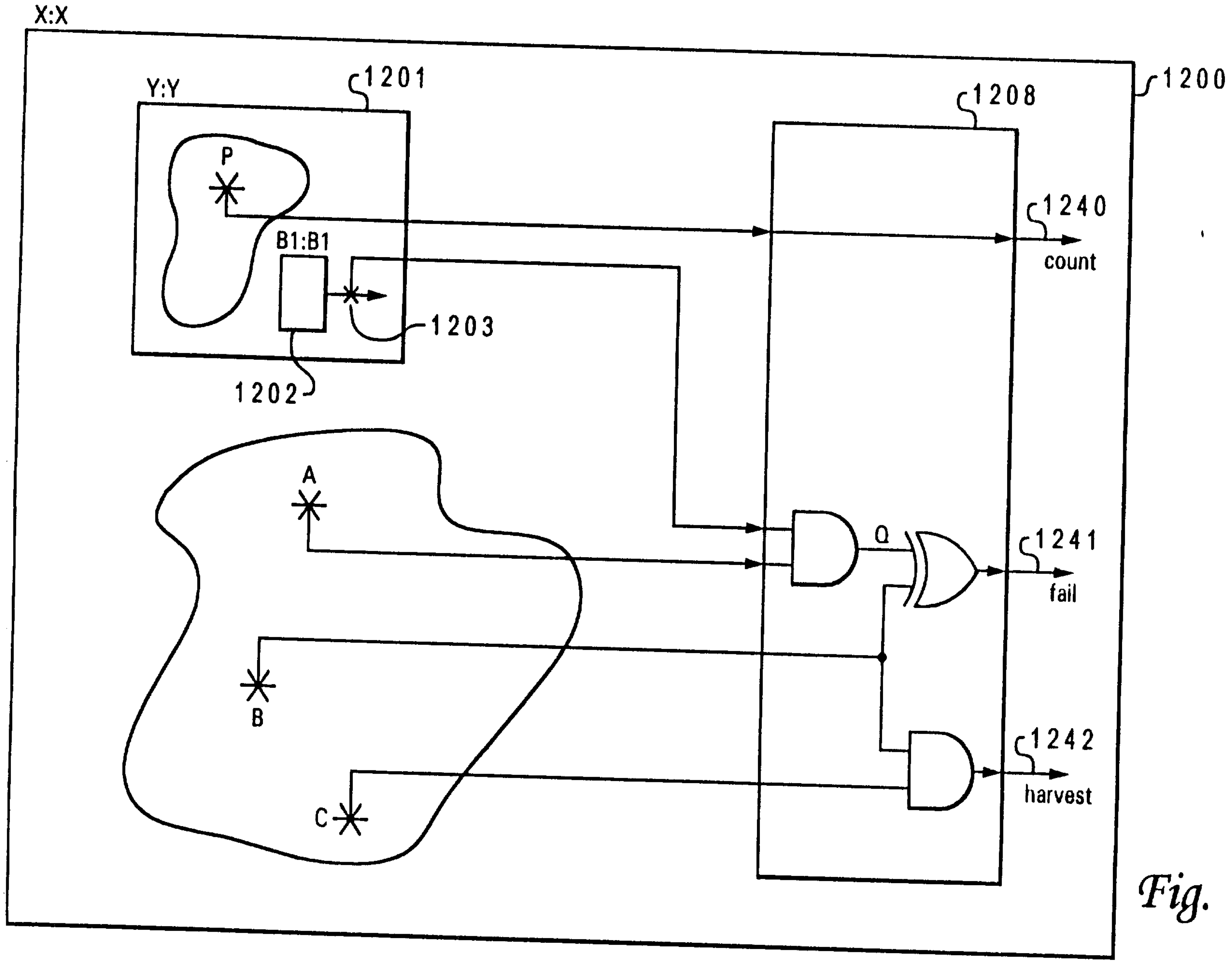
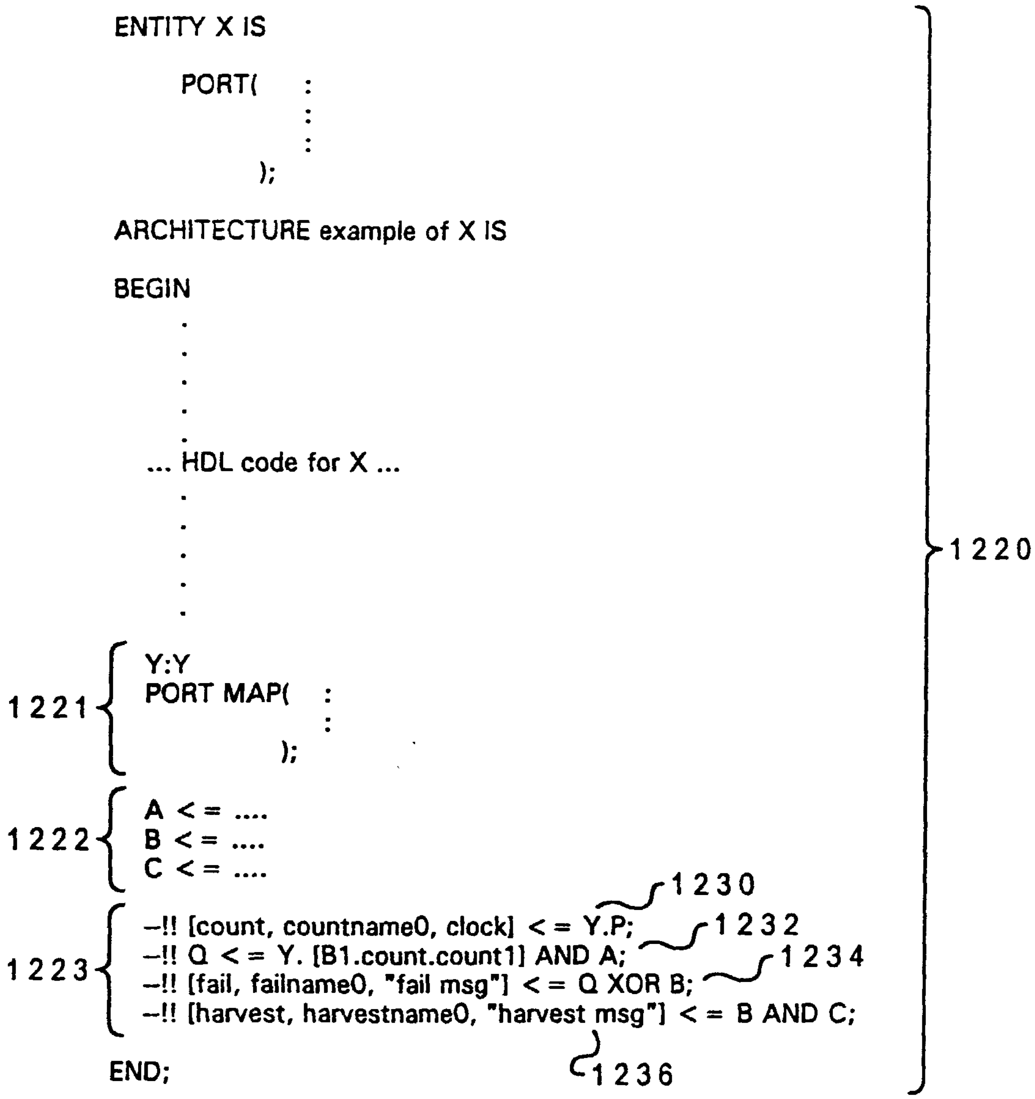

X:X
Y:Y
1201
1200
1208
P
B1:B1
1240
count
1203
1202
A
Q
1241
fail
B
1242
harvest
C

Fig. 12A



```
ENTITY X IS
      PORT(   :
              :
              :
          );
ARCHITECTURE example of X IS
BEGIN
      .
      .
      .
      .
      .
   ... HDL code for X ...
      .
      .
      .
      .
      .
   Y:Y
   PORT MAP(   :
               :
            );
   A < = ....
   B < = ....
   C < = ....
   –!! [count, countname0, clock] < = Y.P;
   –!! Q < = Y. [B1.count.count1] AND A;
   –!! [fail, failname0, "fail msg"] < = Q XOR B;
   –!! [harvest, harvestname0, "harvest msg"] < = B AND C;
END;
```

*Fig. 12B*

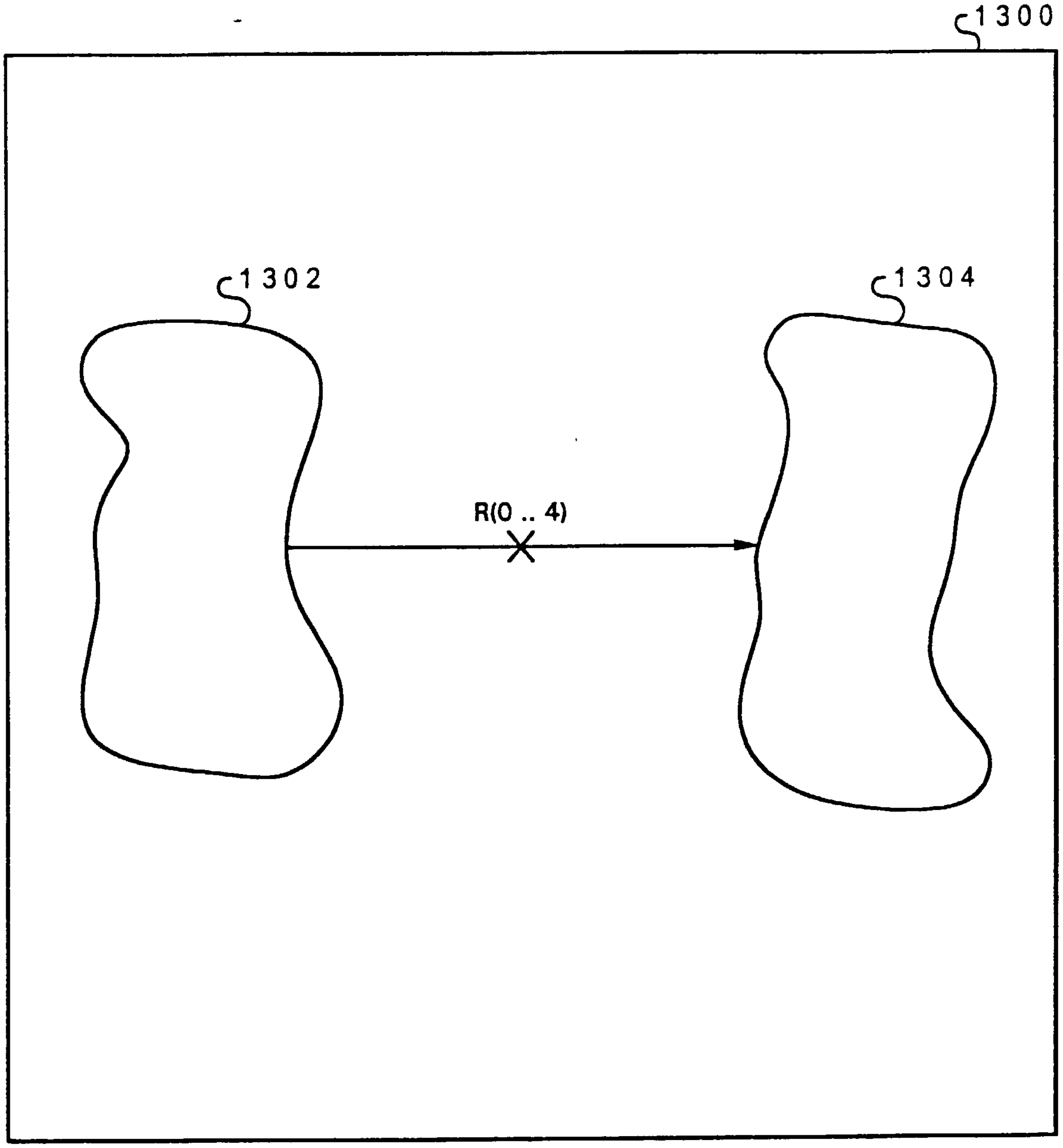


*Fig. 13A*

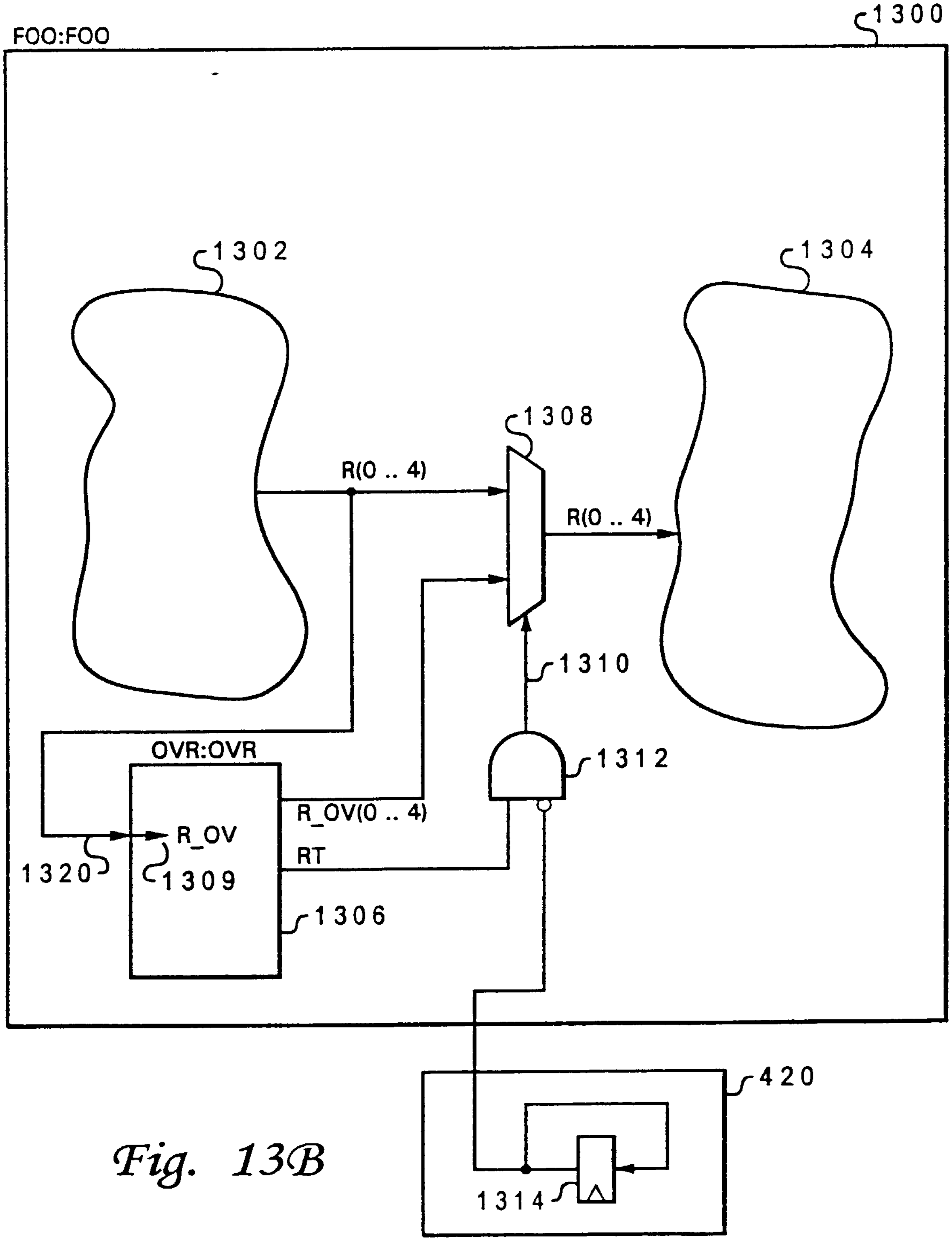
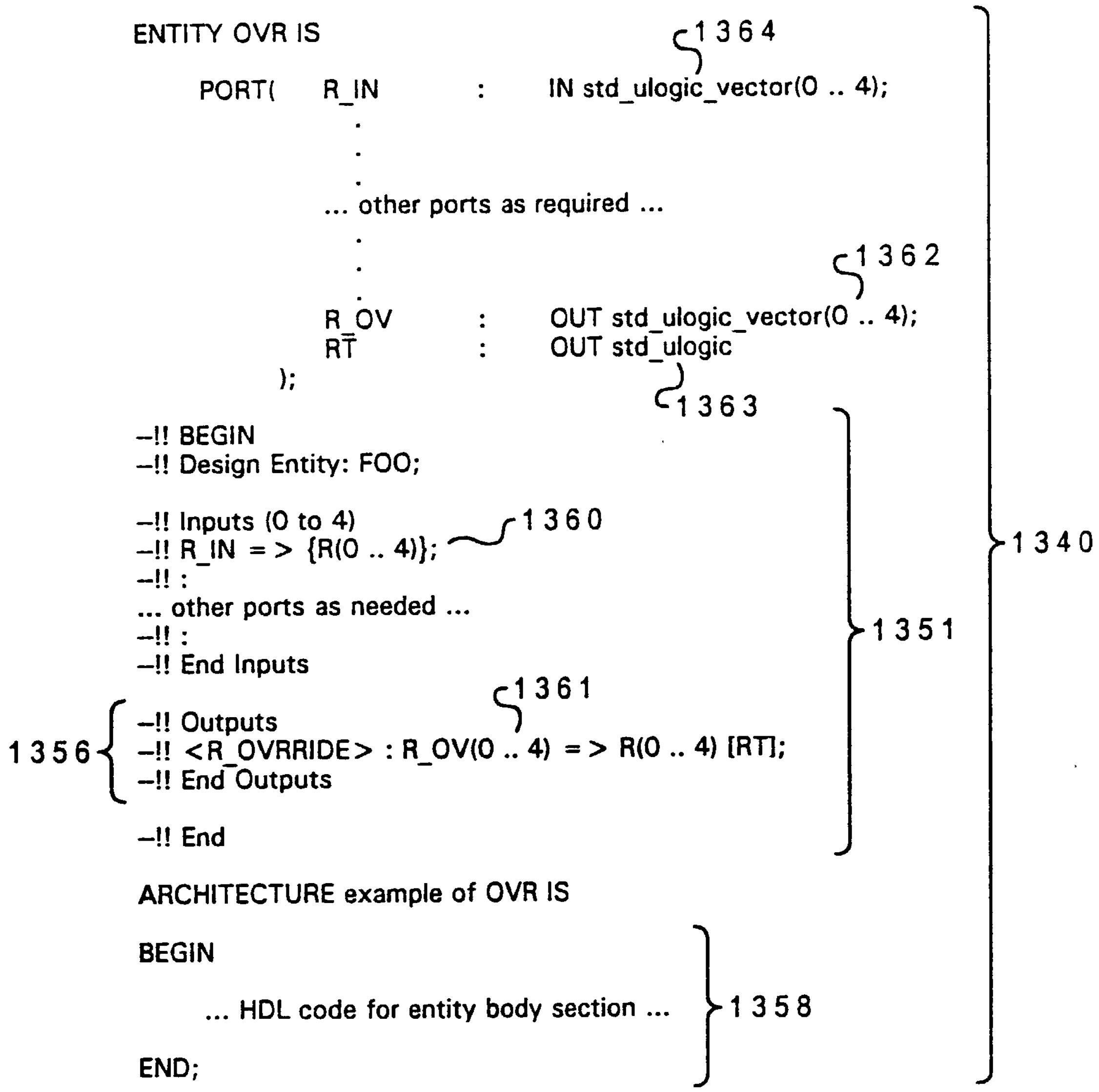
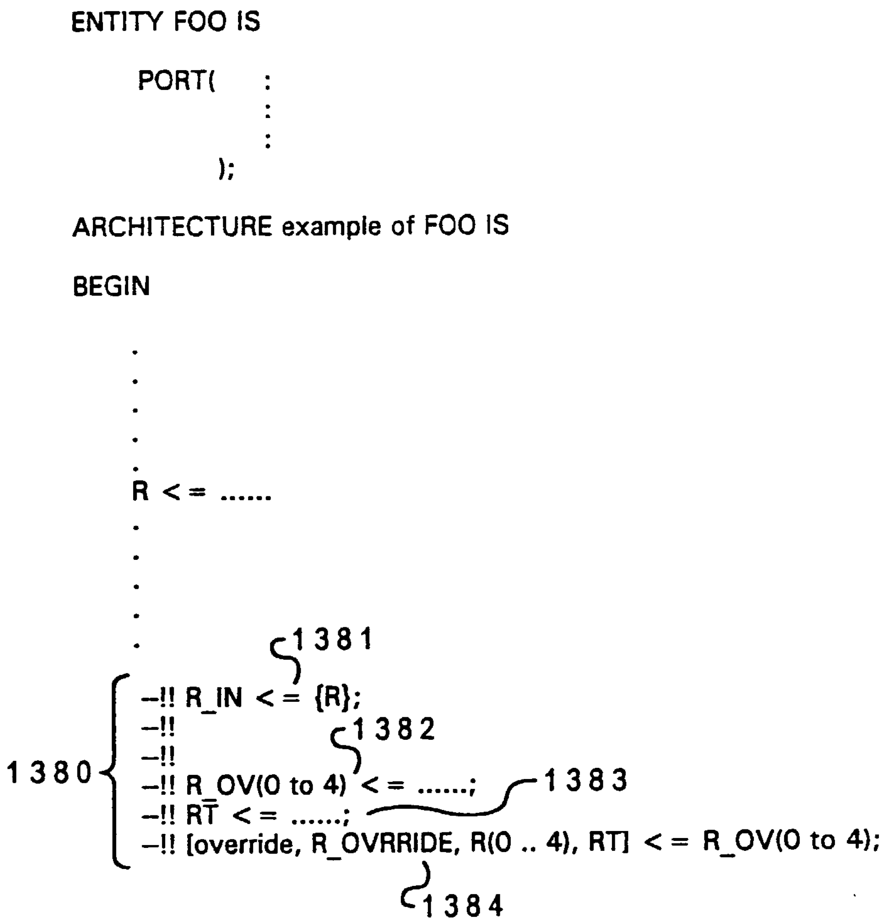

1300
FOO:FOO
1302
1304
1308
R(0 .. 4)
R(0 .. 4)
1310
1312
OVR:OVR
R_OV(0 .. 4)
R_OV
RT
1320
1309
1306
420
1314

*Fig. 13B*



```
ENTITY OVR IS

     PORT(    R_IN          :     IN std_ulogic_vector(0 .. 4);
              .
              .
              .
              ... other ports as required ...
              .
              .
              .
              R_OV          :     OUT std_ulogic_vector(0 .. 4);
              RT            :     OUT std_ulogic
          );

--!! BEGIN
--!! Design Entity: FOO;

--!! Inputs (0 to 4)
--!! R_IN => {R(0 .. 4)};
--!! :
... other ports as needed ...
--!! :
--!! End Inputs

--!! Outputs
--!! <R_OVRRIDE> : R_OV(0 .. 4) => R(0 .. 4) [RT];
--!! End Outputs

--!! End

ARCHITECTURE example of OVR IS

BEGIN

      ... HDL code for entity body section ...

END;
```

*Fig. 13C*



```
ENTITY FOO IS

      PORT(    :
               :
               :
            );

ARCHITECTURE example of FOO IS

BEGIN

      .
      .
      .
      .
      .
      R <= ......
      .
      .
      .
      .
      .
      --!! R_IN <= {R};
      --!!
      --!!
      --!! R_OV(0 to 4) <= ......;
      --!! RT <= ......;
      --!! [override, R_OVRRIDE, R(0 .. 4), RT] <= R_OV(0 to 4);
```

*Fig. 13D*

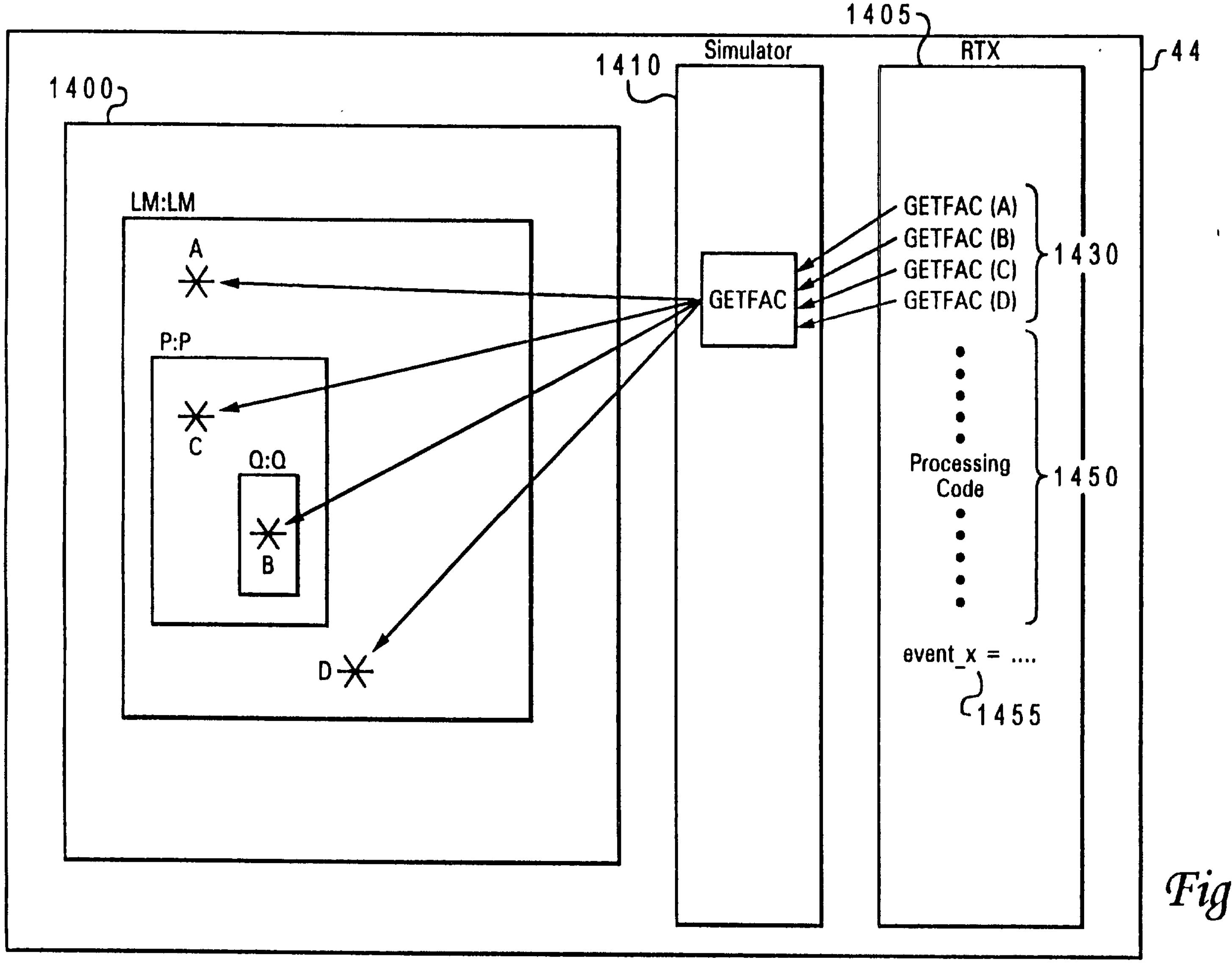

44
1405
RTX
1410
Simulator
1400
LM:LM
A
P:P
C
Q:Q
B
D
GETFAC
GETFAC (A)
GETFAC (B)
GETFAC (C)
GETFAC (D)
1430
Processing
Code
1450
event_x = ....
1455

*Fig. 14A*

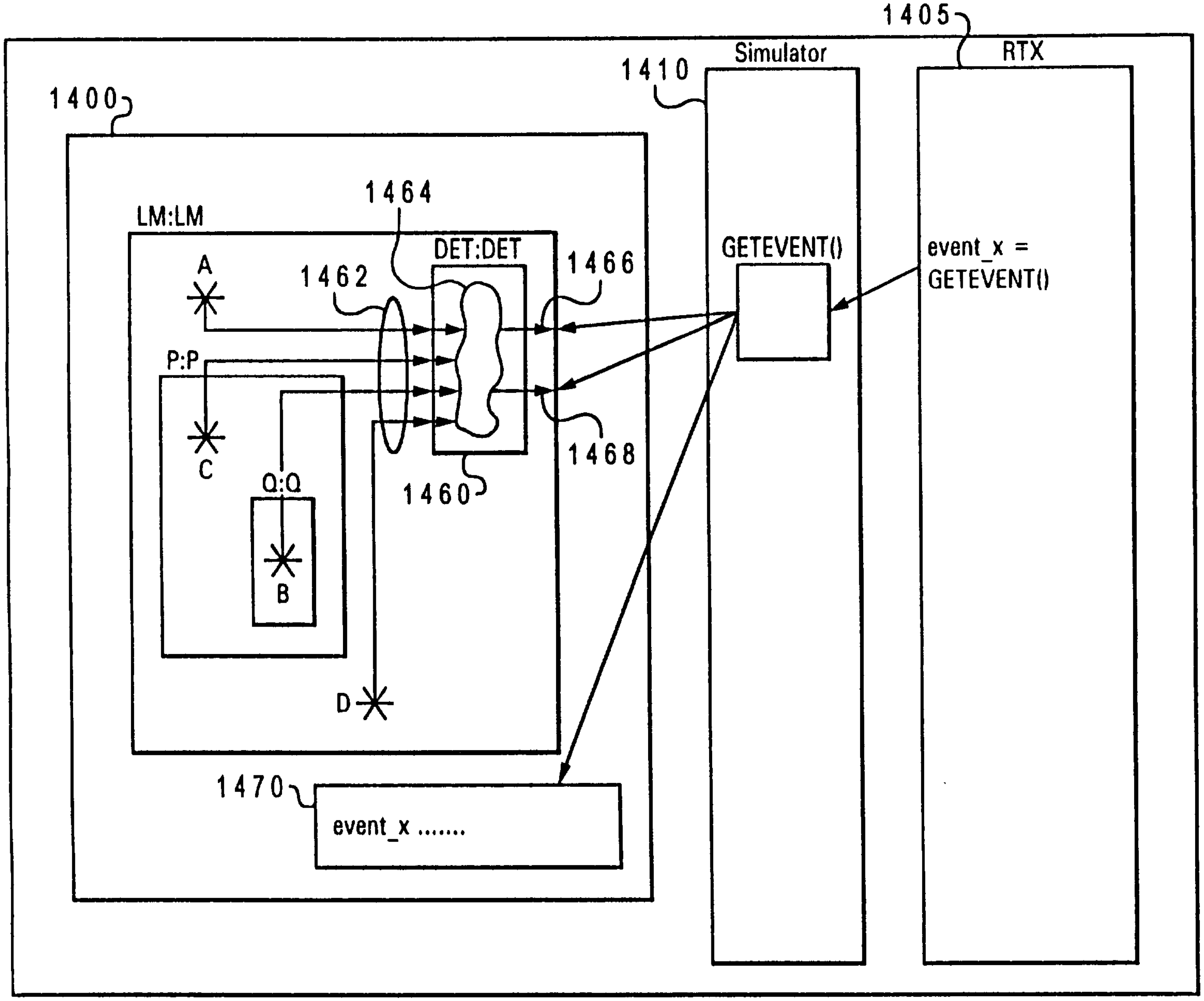
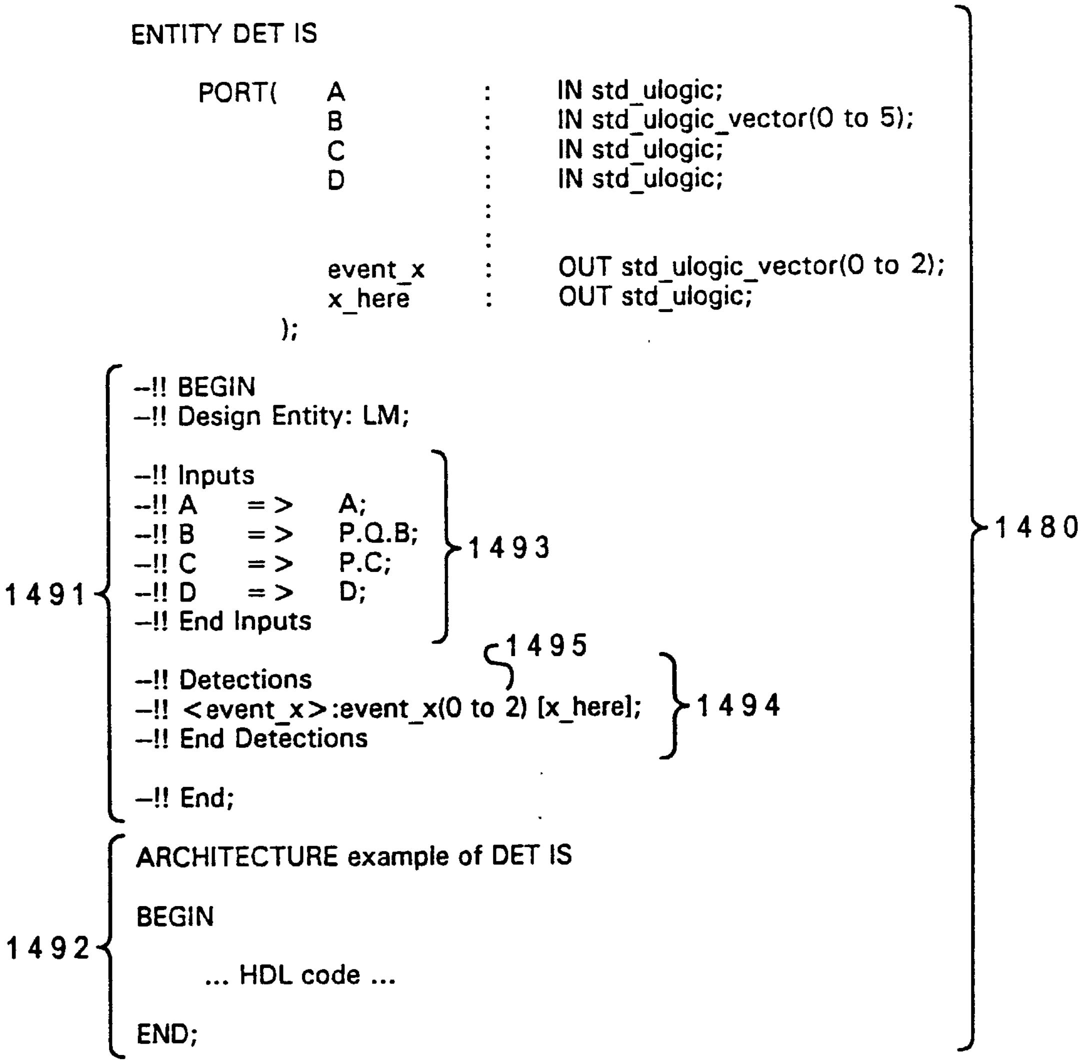
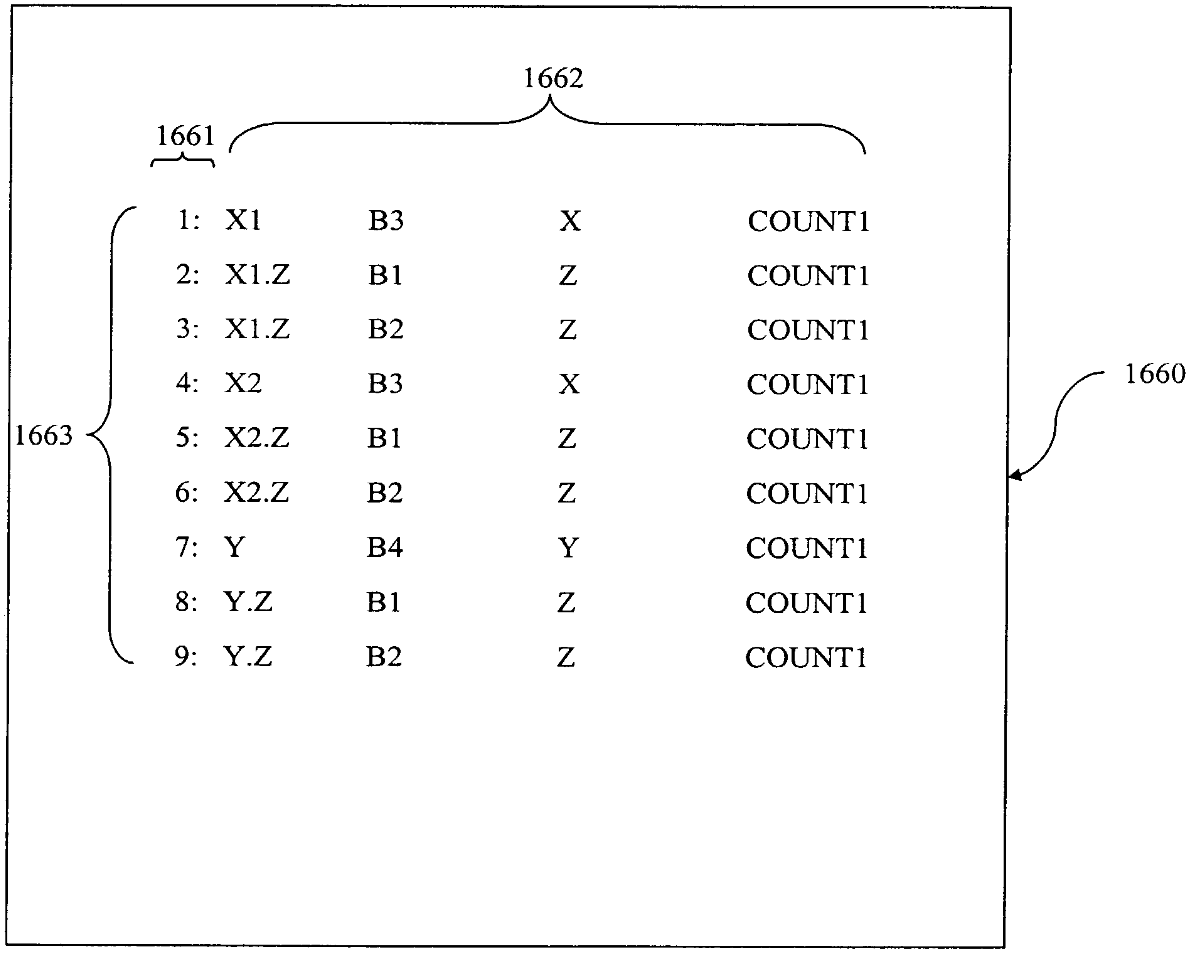

1405
1410
Simulator
RTX
1400
1464
LM:LM
DET:DET
1466
GETEVENT()
event_x =
GETEVENT()
A
1462
P:P
C
Q:Q
1460
1468
B
D
1470
event_x .......

Fig. 14B



```
ENTITY DET IS

     PORT(    A           :     IN std_ulogic;
              B           :     IN std_ulogic_vector(0 to 5);
              C           :     IN std_ulogic;
              D           :     IN std_ulogic;
                          :
                          :
              event_x     :     OUT std_ulogic_vector(0 to 2);
              x_here      :     OUT std_ulogic;
         );

--!! BEGIN
--!! Design Entity: LM;

--!! Inputs
--!! A     =>    A;
--!! B     =>    P.Q.B;
--!! C     =>    P.C;
--!! D     =>    D;
--!! End Inputs

--!! Detections
--!! <event_x>:event_x(0 to 2) [x_here];
--!! End Detections

--!! End;

ARCHITECTURE example of DET IS

BEGIN

     ... HDL code ...

END;
```

*Fig. 14C*



| | | | | |
|---|---|---|---|---|
| 1: | X1 | B3 | X | COUNT1 |
| 2: | X1.Z | B1 | Z | COUNT1 |
| 3: | X1.Z | B2 | Z | COUNT1 |
| 4: | X2 | B3 | X | COUNT1 |
| 5: | X2.Z | B1 | Z | COUNT1 |
| 6: | X2.Z | B2 | Z | COUNT1 |
| 7: | Y | B4 | Y | COUNT1 |
| 8: | Y.Z | B1 | Z | COUNT1 |
| 9: | Y.Z | B2 | Z | COUNT1 |

FIG. 15

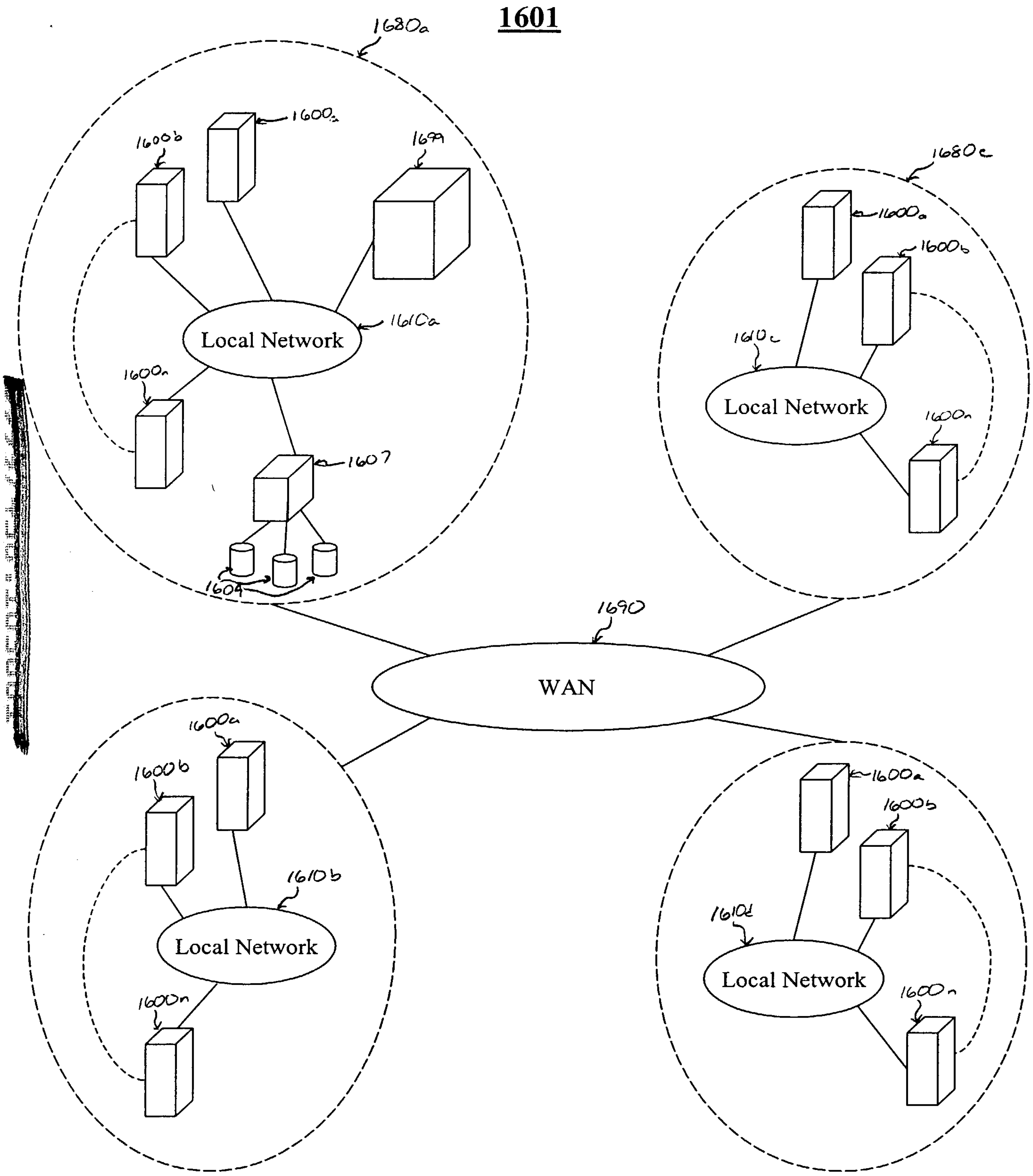

1601
1680a
1600a
1600b
1699
Local Network
1610a
1600n
1607
1604
1680c
1600a
1600b
1610c
Local Network
1600n
1690
WAN
1600a
1600b
1610b
Local Network
1600n
1600a
1600b
1610d
Local Network
1600n

FIG. 16B

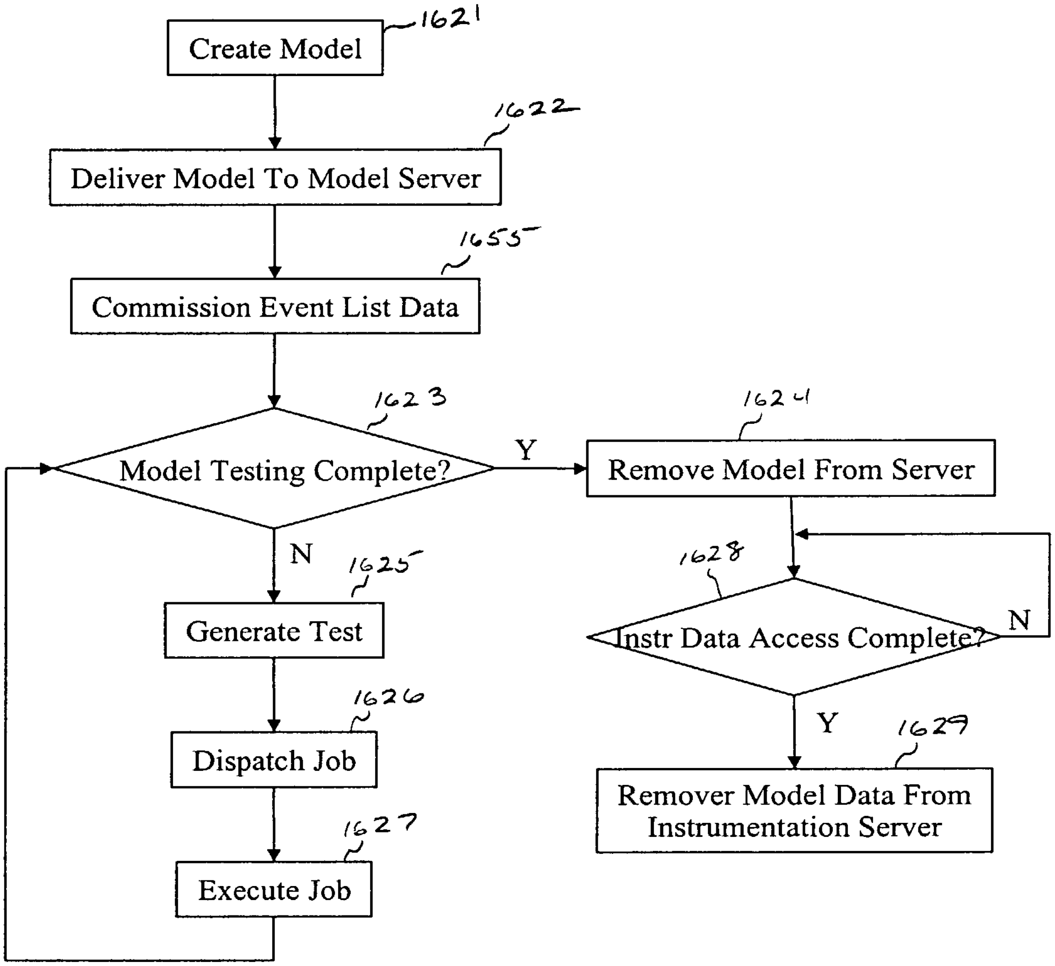

Create Model
1621
Deliver Model To Model Server
1622
Commission Event List Data
1655
Model Testing Complete?
1623
Y
Remove Model From Server
1624
N
Generate Test
1625
Dispatch Job
1626
Execute Job
1627
Instr Data Access Complete?
1628
N
Y
Remover Model Data From Instrumentation Server
1629

FIG. 16C

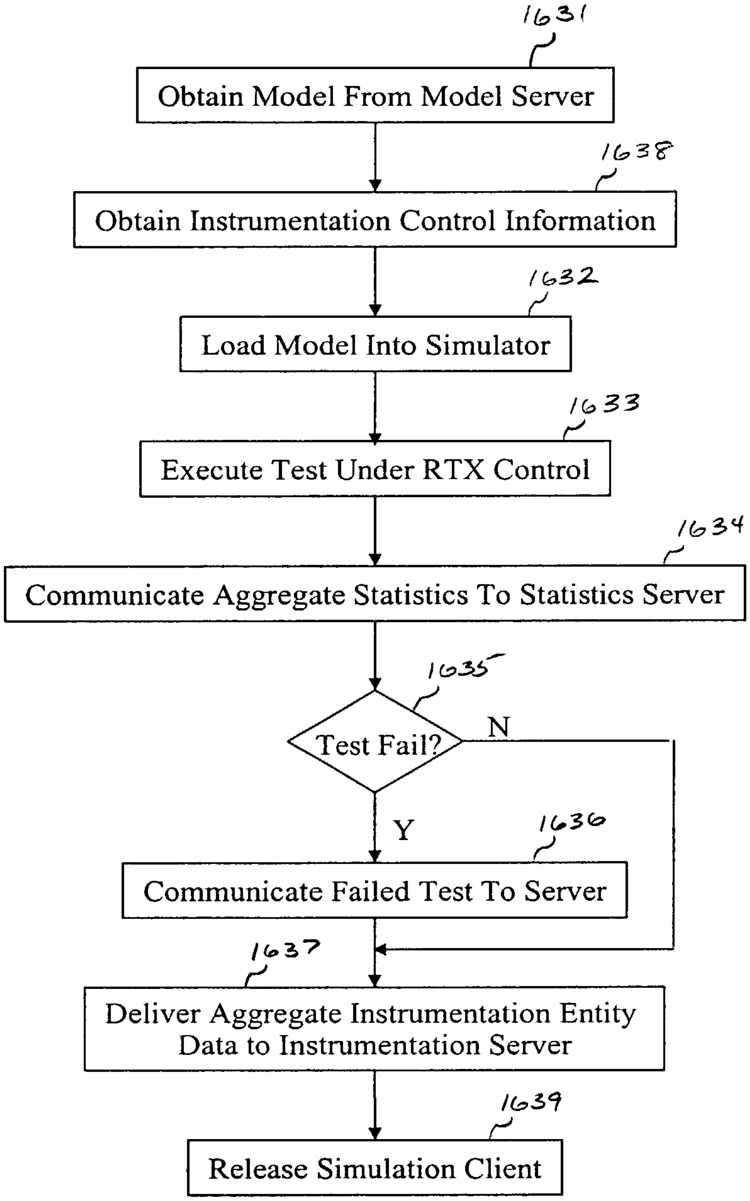

1631
Obtain Model From Model Server
1638
Obtain Instrumentation Control Information
1632
Load Model Into Simulator
1633
Execute Test Under RTX Control
1634
Communicate Aggregate Statistics To Statistics Server
1635
Test Fail?
N
Y
1636
Communicate Failed Test To Server
1637
Deliver Aggregate Instrumentation Entity Data to Instrumentation Server
1639
Release Simulation Client

FIG. 16D

[redacted]

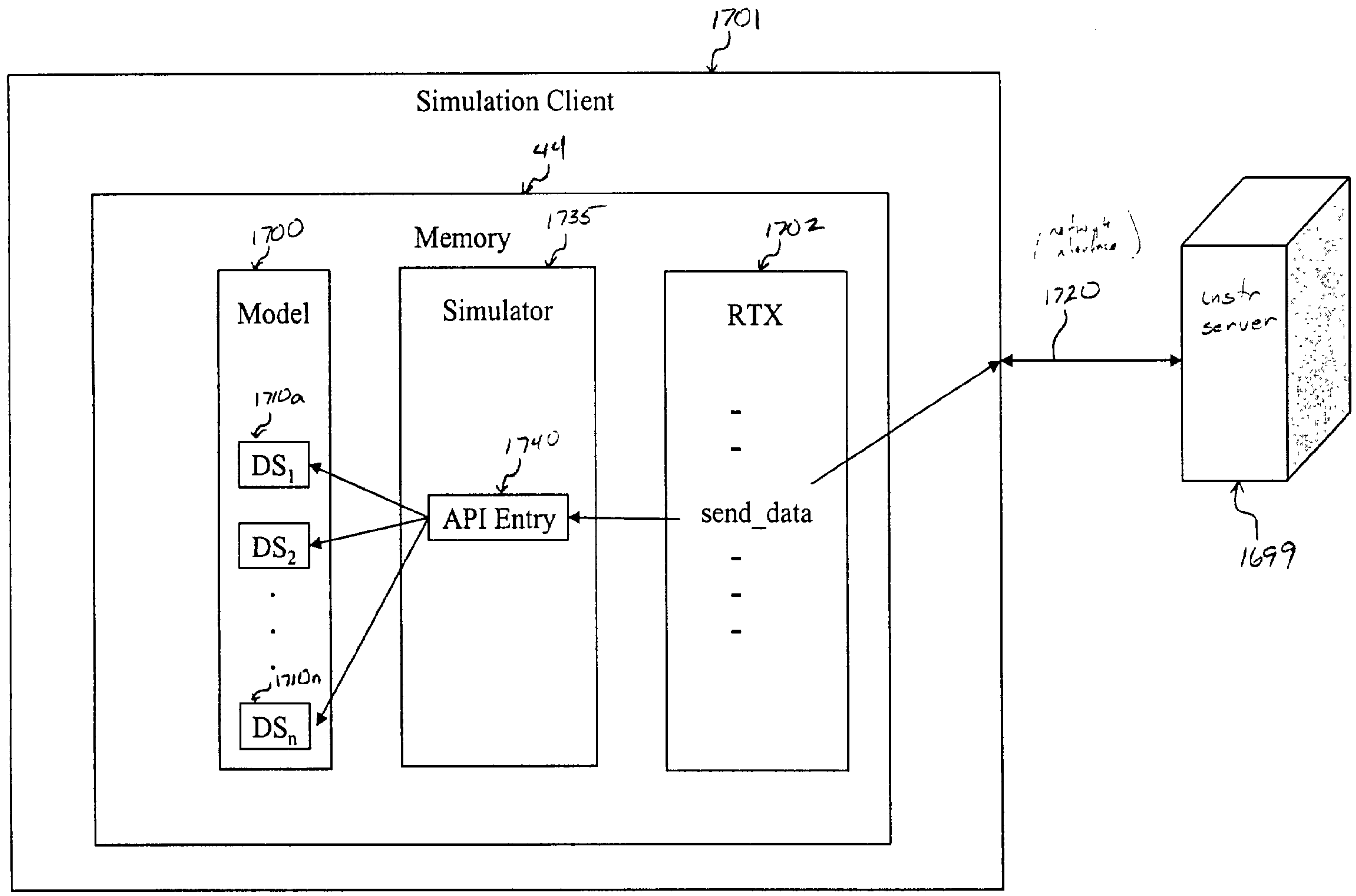


FIG. 17A

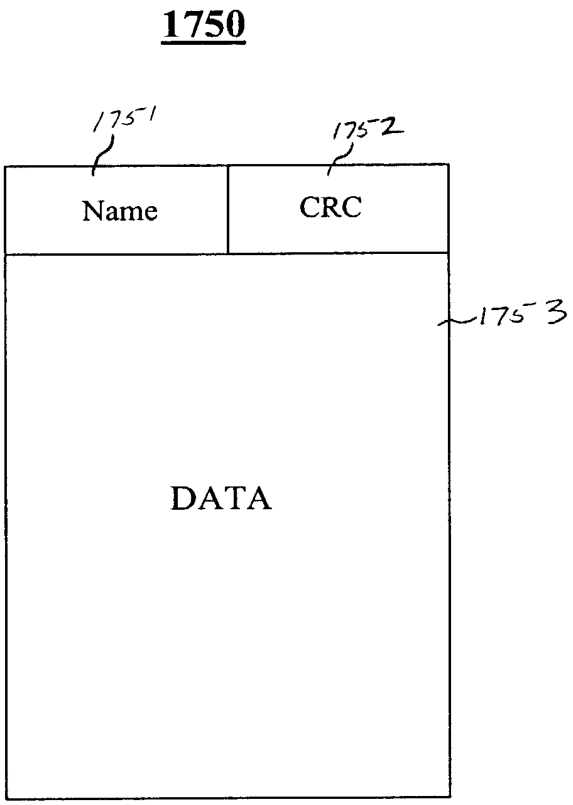


FIG. 17B

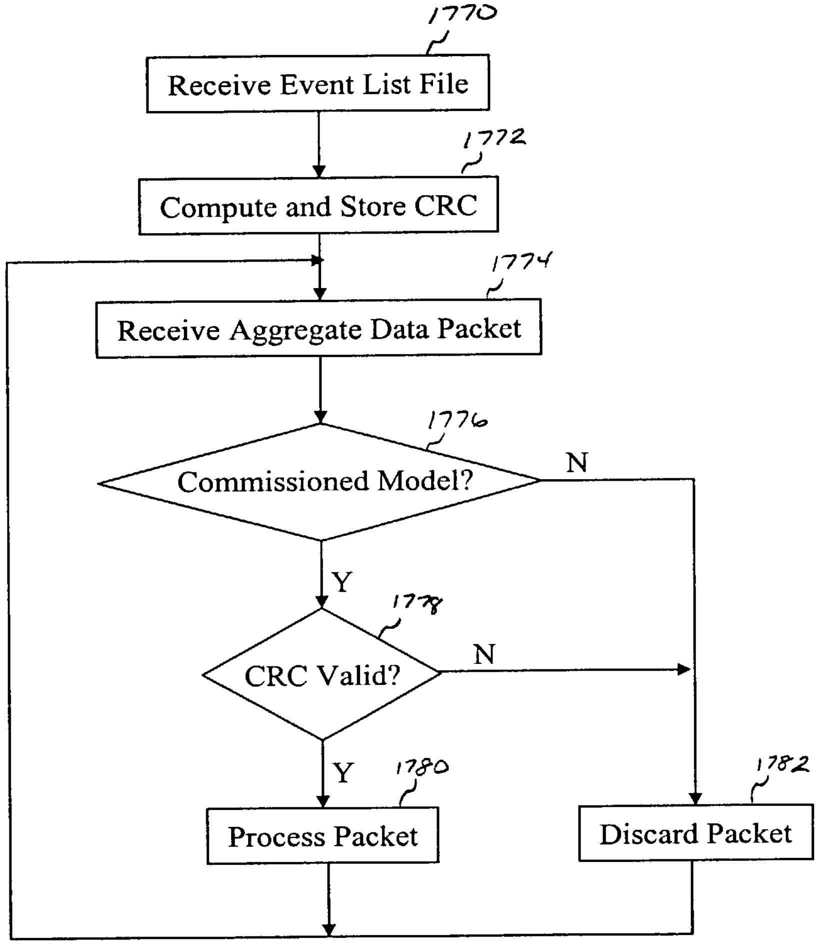

1770
Receive Event List File
1772
Compute and Store CRC
1774
Receive Aggregate Data Packet
1776
Commissioned Model?
N
Y
1778
CRC Valid?
N
Y
1780
Process Packet
1782
Discard Packet

FIG. 17C

[redacted]

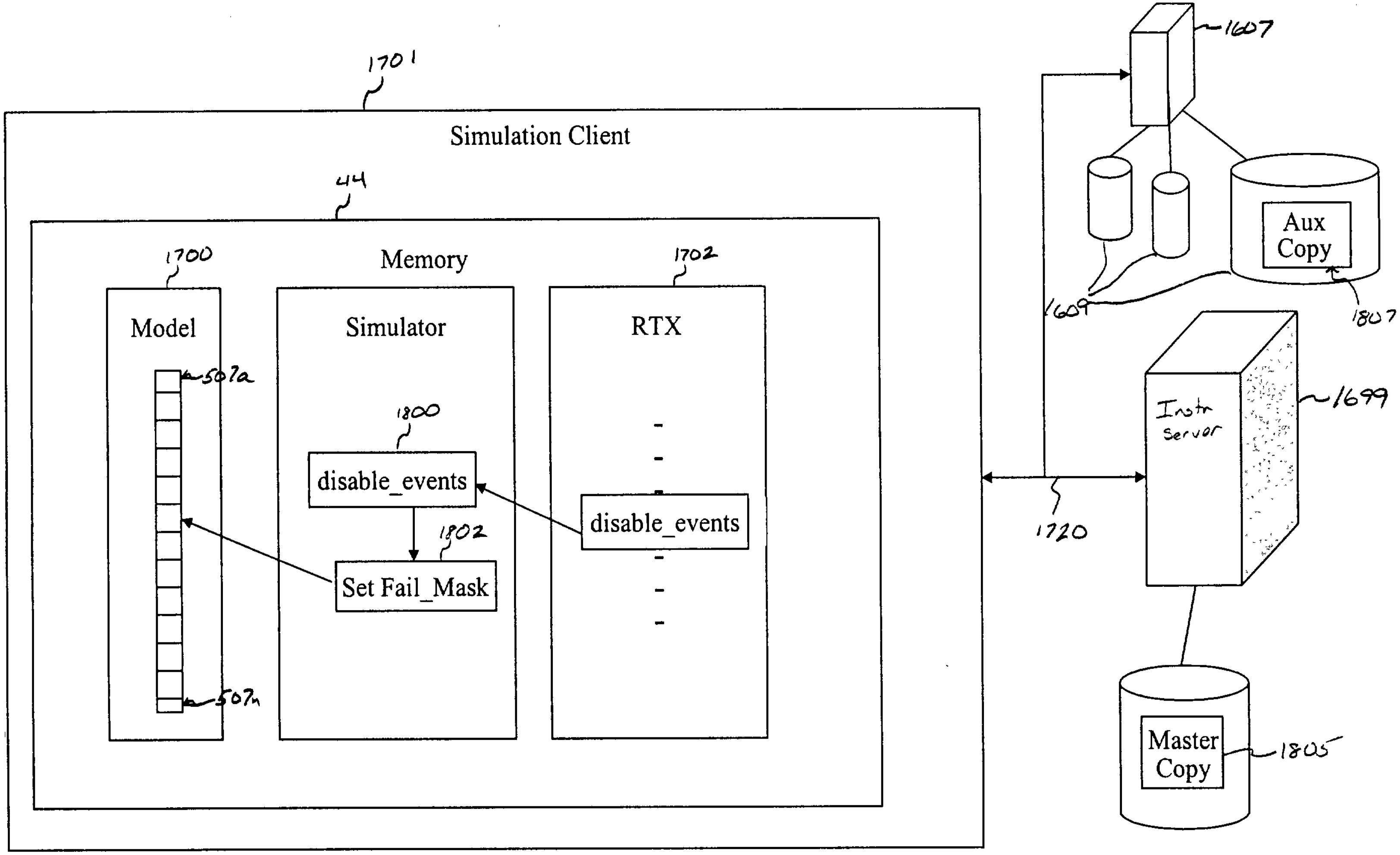


FIG. 18A

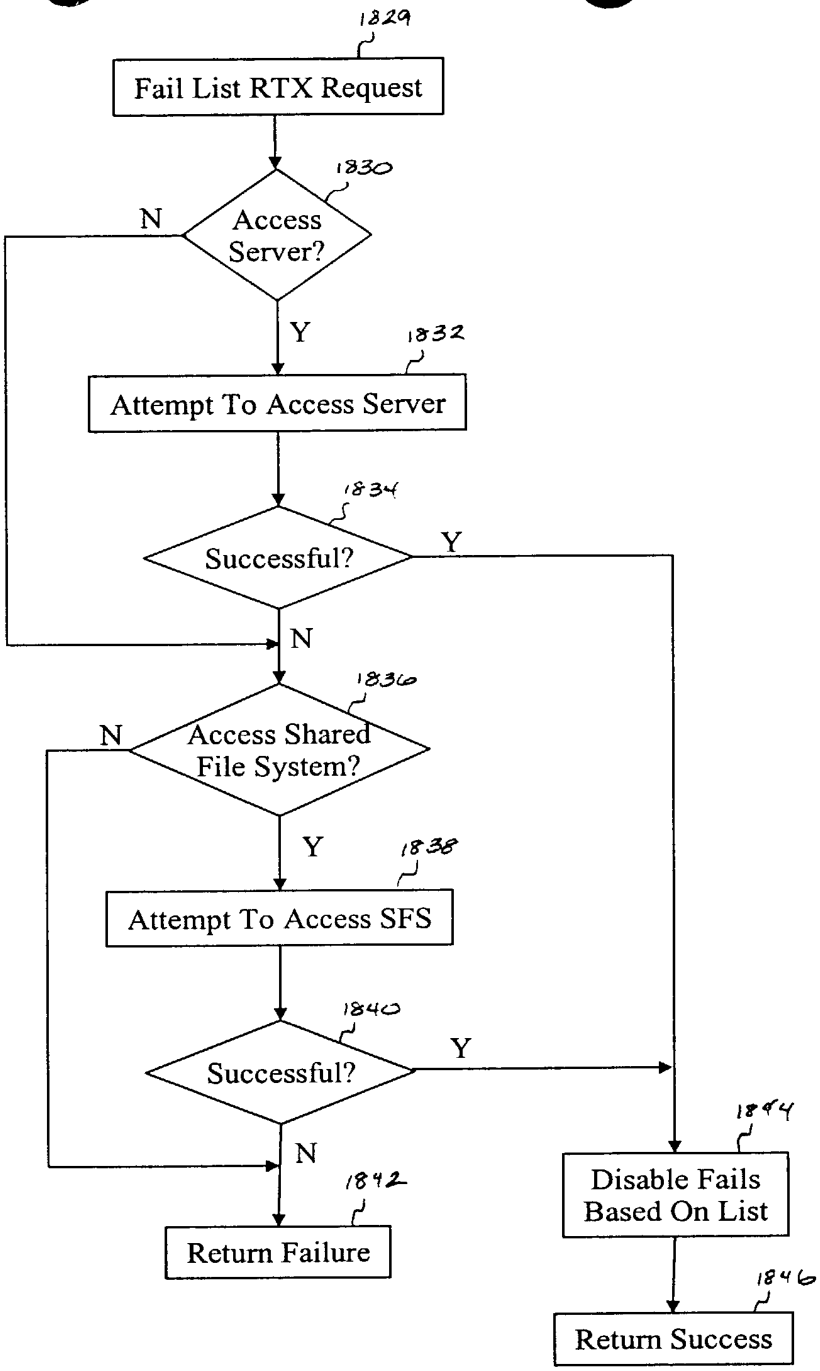

1829
Fail List RTX Request
1830
Access Server?
N
Y
1832
Attempt To Access Server
1834
Successful?
Y
N
1836
Access Shared File System?
N
Y
1838
Attempt To Access SFS
1840
Successful?
Y
N
1842
Return Failure
1844
Disable Fails Based On List
1846
Return Success

FIG. 18B

[redacted]

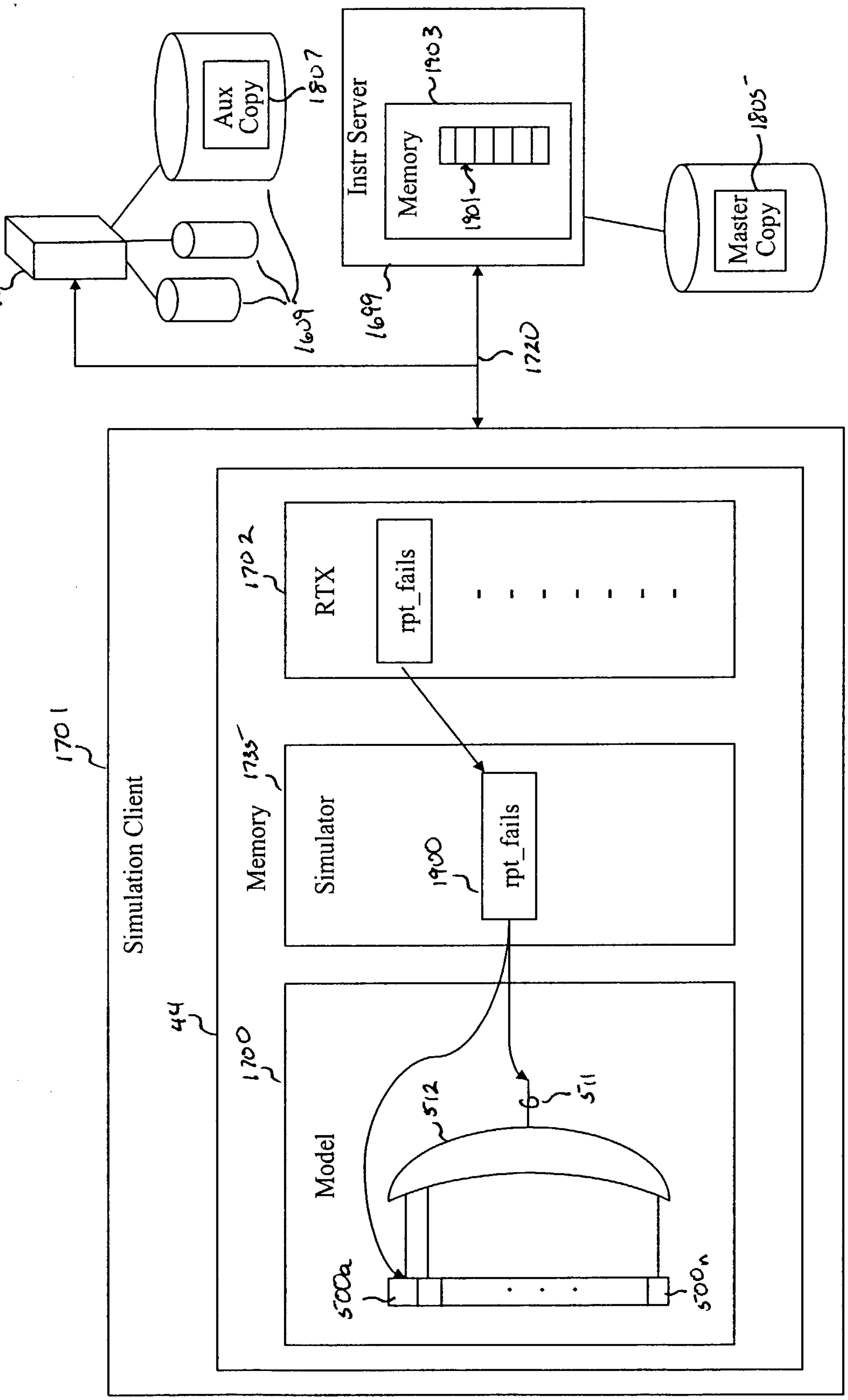


FIG. 19A

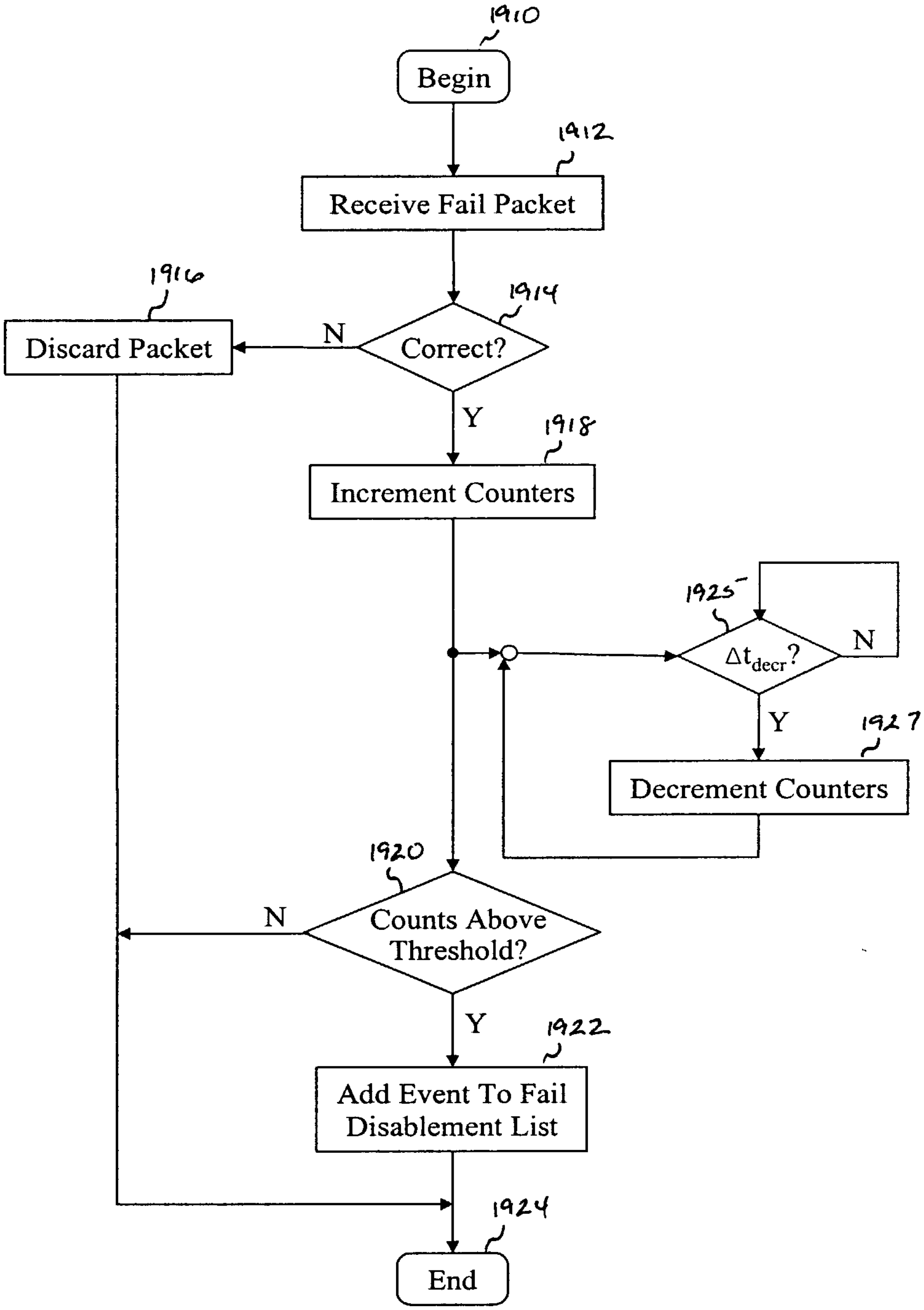

1910
Begin
1912
Receive Fail Packet
1914
Correct?
N
1916
Discard Packet
Y
1918
Increment Counters
1925
$\Delta t_{decr}$?
N
Y
1927
Decrement Counters
1920
Counts Above Threshold?
N
Y
1922
Add Event To Fail Disablement List
1924
End

FIG. 19B

[redacted]

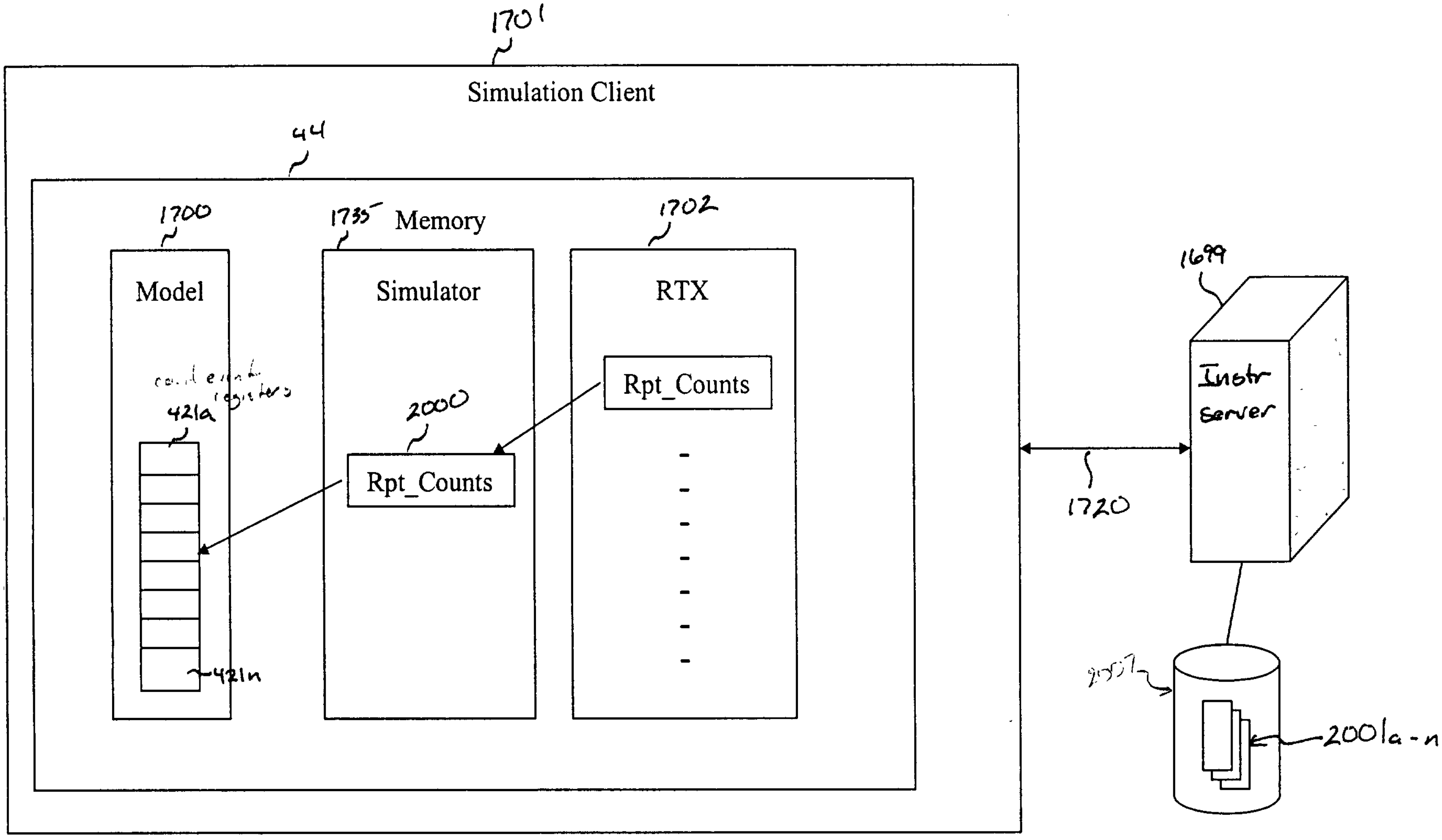


FIG. 20A

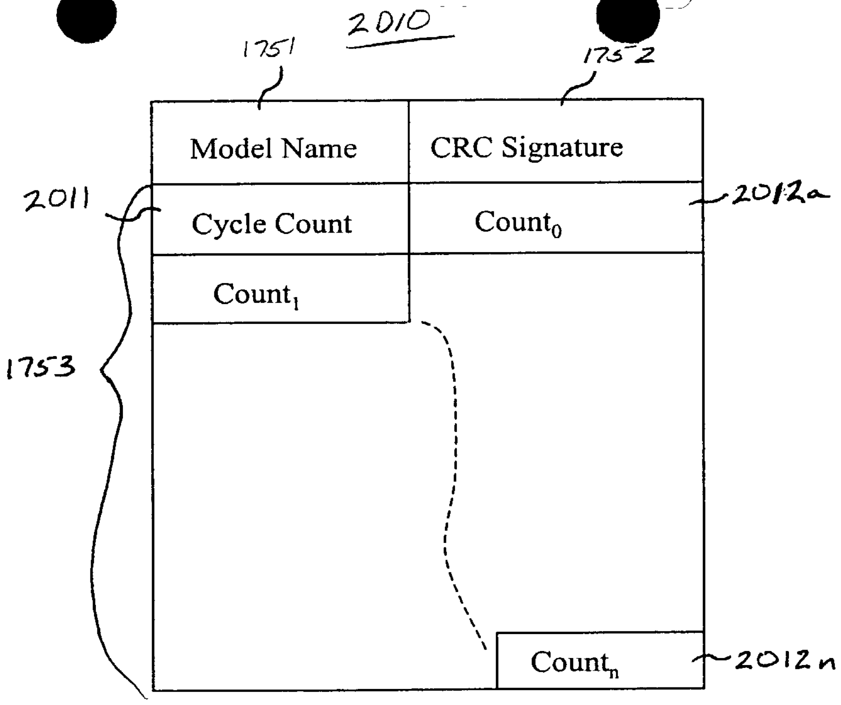


FIG. 20B

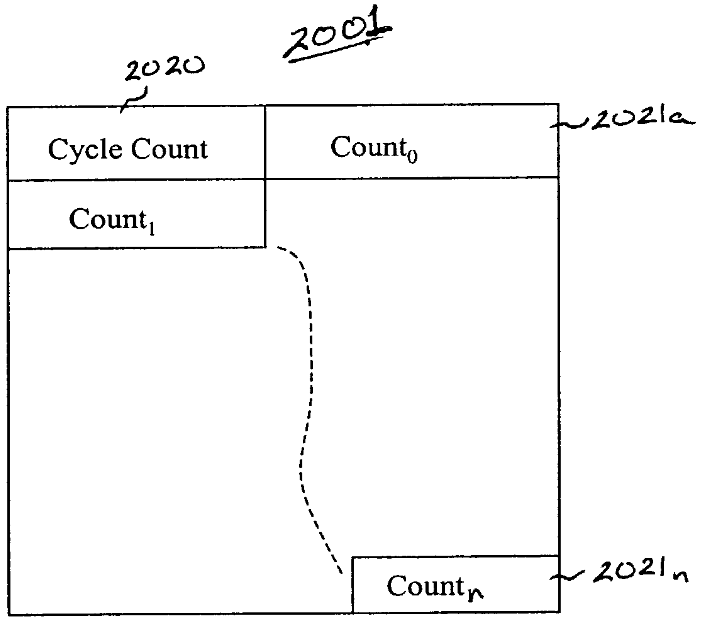


FIG. 20C

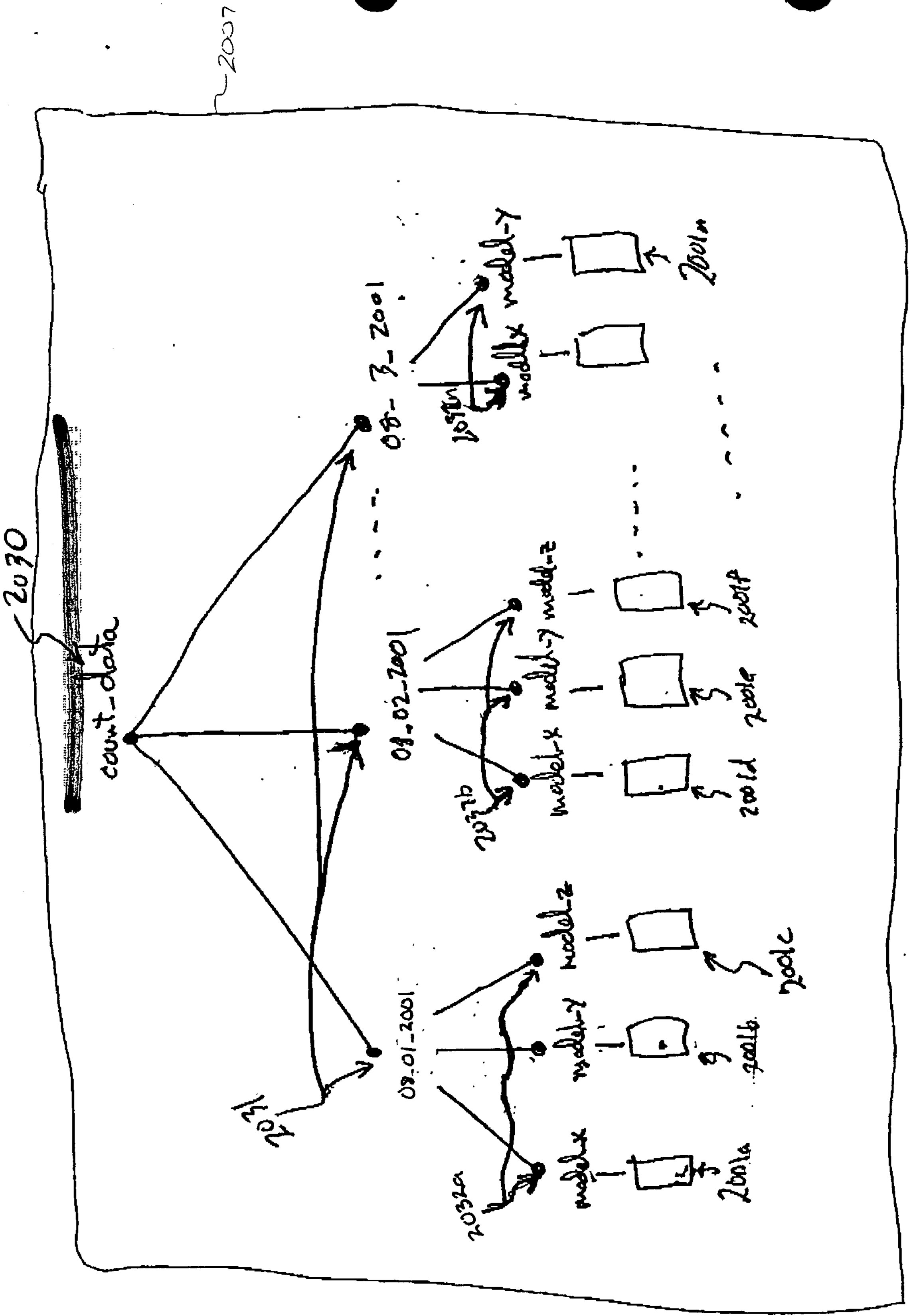

2007
2030
count_data
08_01_2001
08_02_2001
08- 3_2001
model_x
model_y
model_z
model-x
model-y
model-z
model_x
model_y
2001a
2001b
2001c
2001d
2001e
2001f
2001n
2031
2032a
2032b
2032n

FIG. 20D

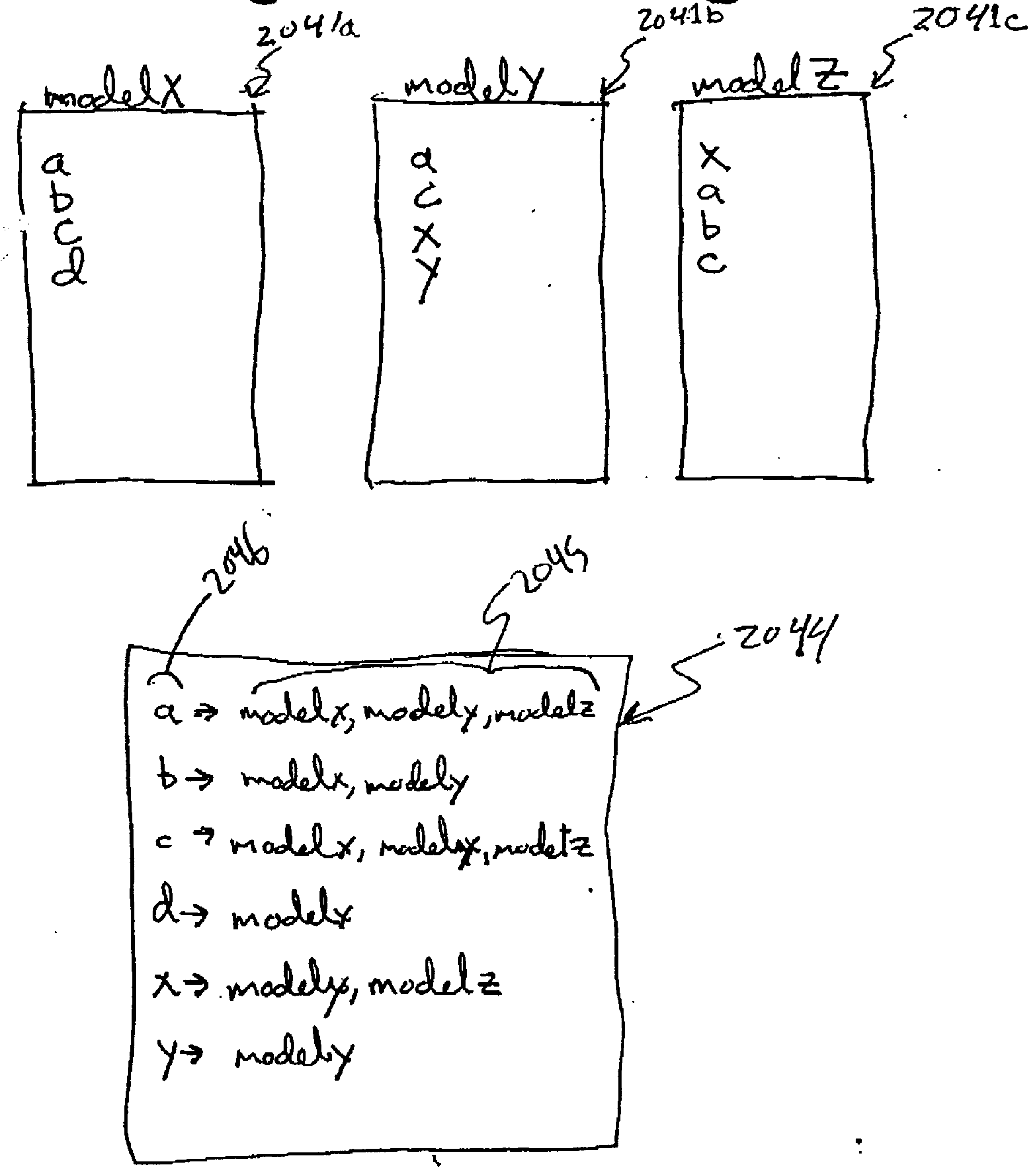

2041a
2041b
2041c
modelX
a
b
c
d
modelY
a
c
x
y
modelZ
x
a
b
c
2046
2045
2044
a → modelx, modely, modelz
b → modelx, modely
c → modelx, modelx, modelz
d → modelx
x → modely, modelz
y → modely

FIG. 20E

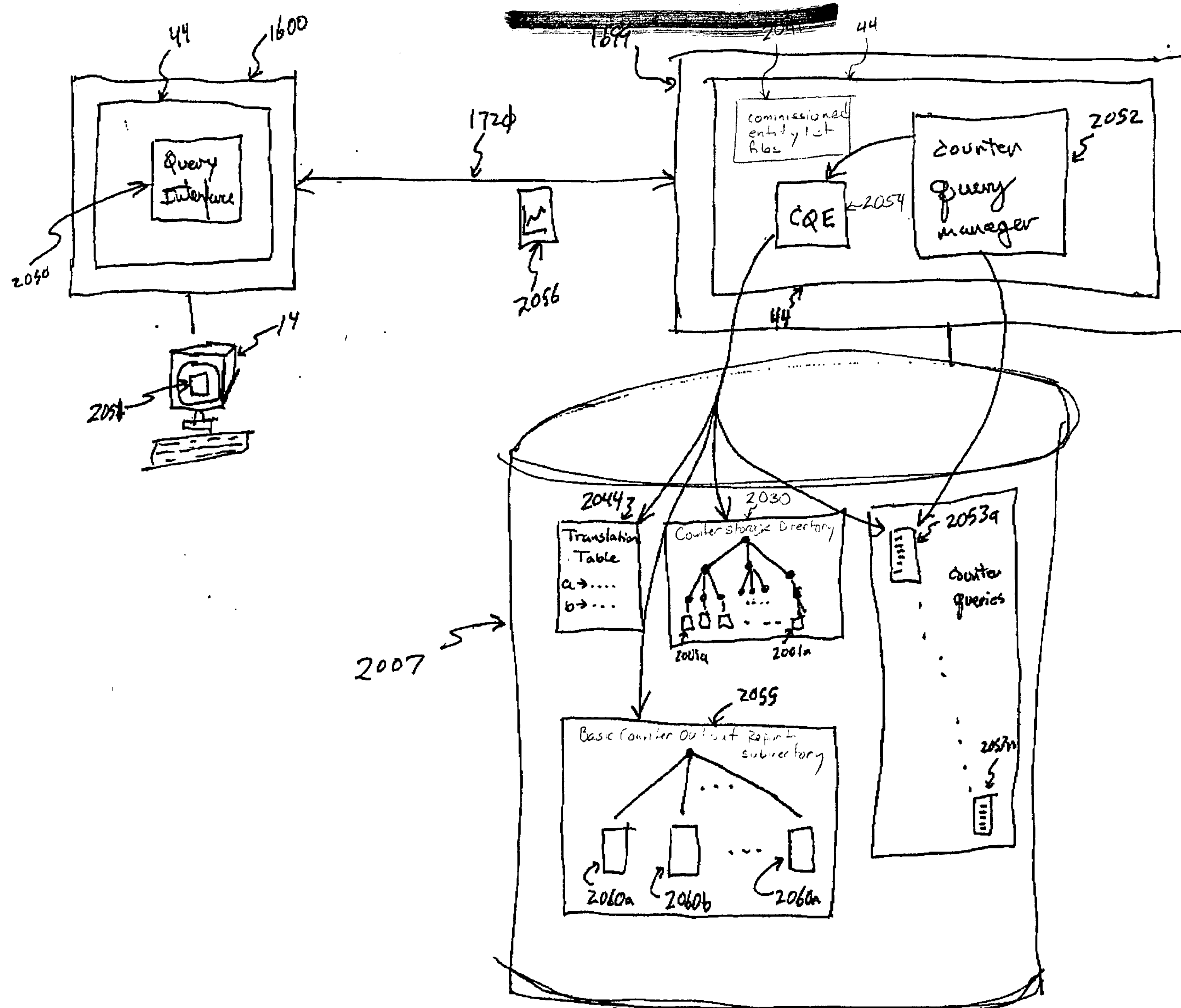
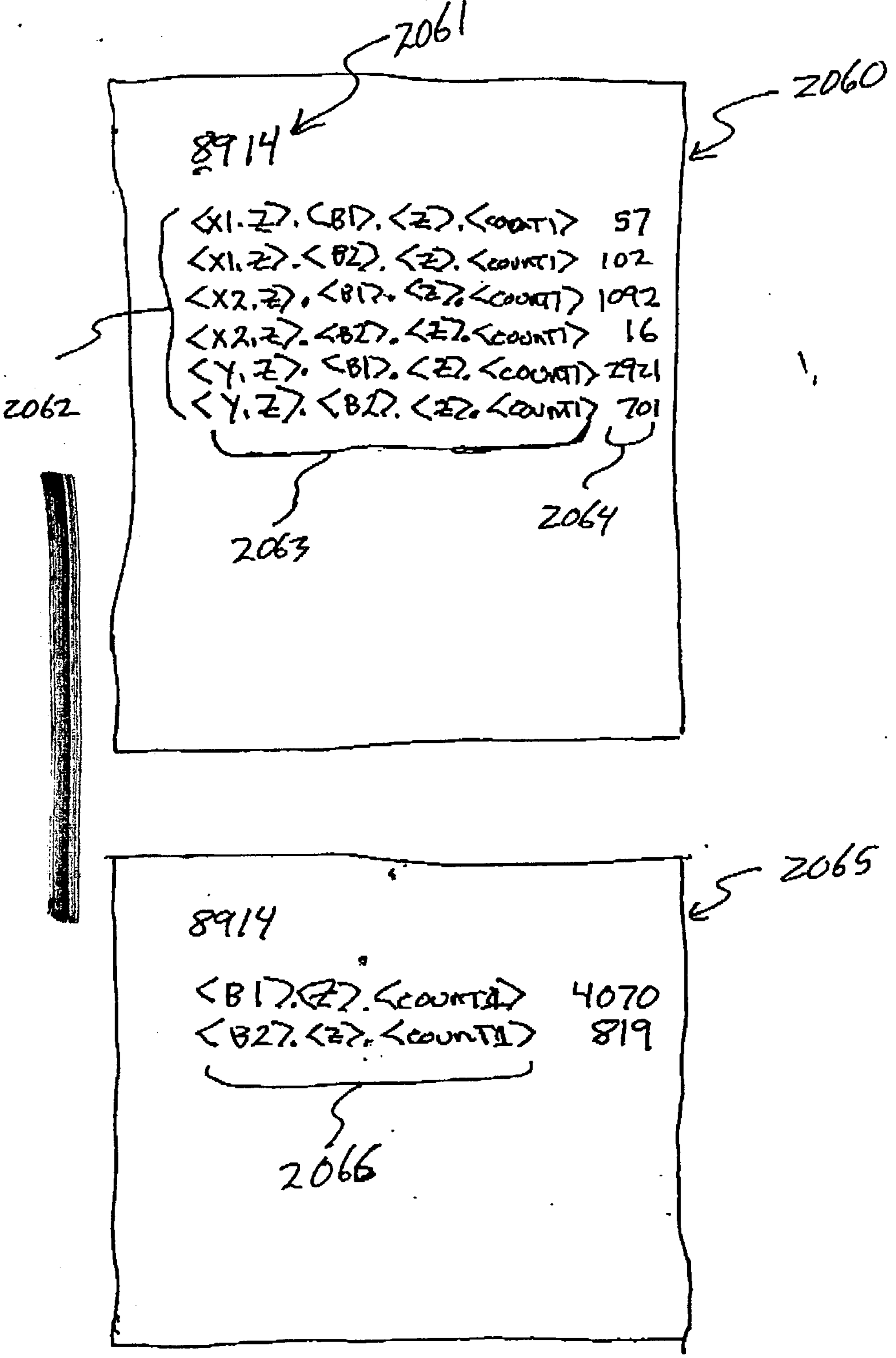

44
1600
Query Interface
2050
1720
2056
14
2058
1699
2041
44
commissioned entity list files
2052
Counter Query manager
CQE
2054
44
2007
2044
Translation Table
a→....
b→...
2030
Counter Storage Directory
2061q
2061n
2053a
counter queries
2053n
2055
Basic Counter Output Report subdirectory
2060a
2060b
2060n

FIG 20F



8914

<X1.Z>.<B1>.<Z>.<COUNT1> 57
<X1,Z>.<B2>.<Z>.<COUNT1> 102
<X2,Z>.<B1>.<Z>.<COUNT1> 1092
<X2,Z>.<B2>.<Z>.<COUNT1> 16
<Y,Z>.<B1>.<Z>.<COUNT1> 2921
<Y,Z>.<B2>.<Z>.<COUNT1> 701

8914

<B1>.<Z>.<COUNT1> 4070
<B2>.<Z>.<COUNT1> 819

FIG. 206

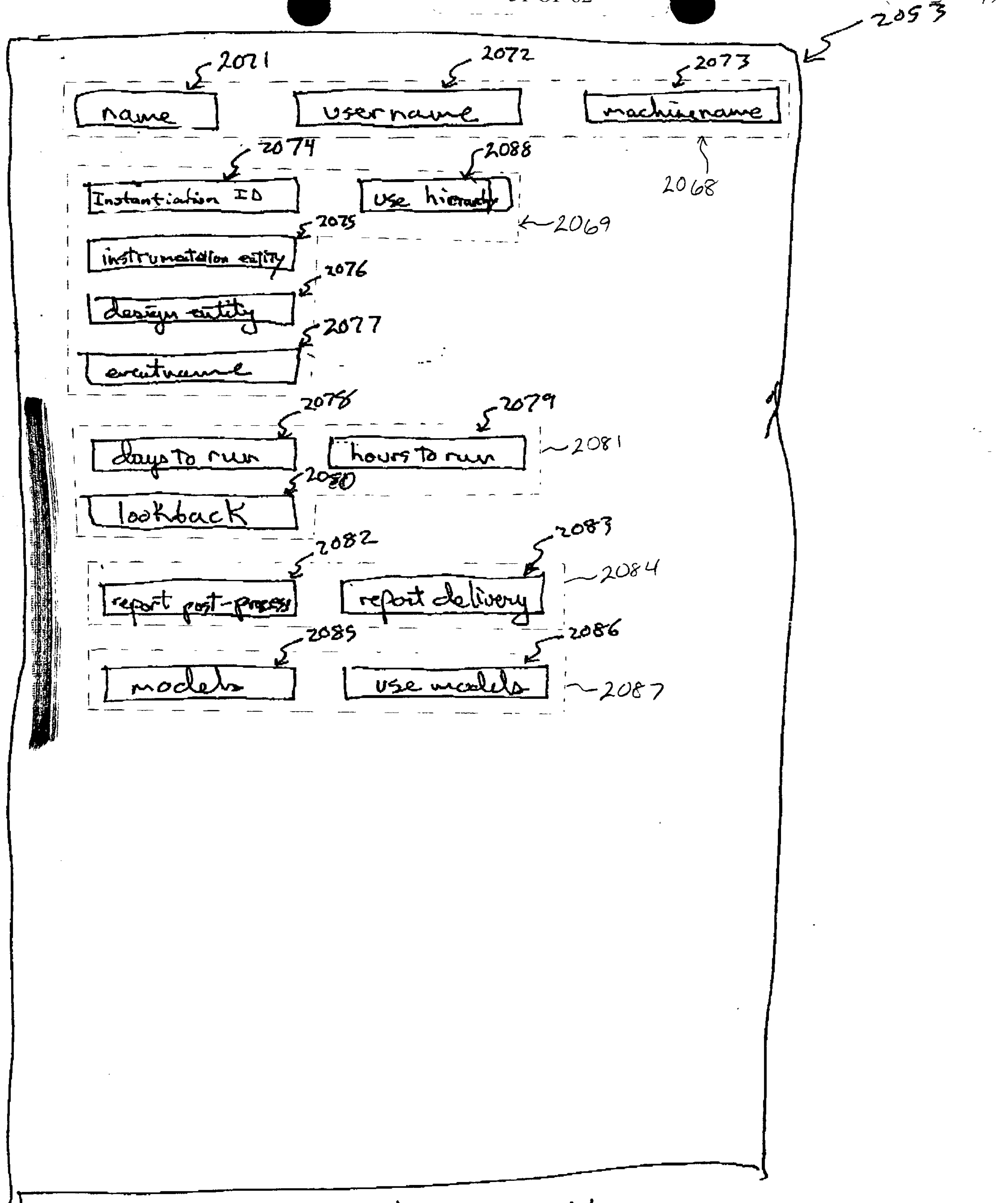

query
2053
2071
name
2072
username
2073
machinename
2068
2074
Instantiation ID
2088
use hierarchy
2069
2075
instrumentation entity
2076
design entity
2077
eventname
2078
days to run
2079
hours to run
2081
2080
lookback
2082
report post-process
2083
report delivery
2084
2085
models
2086
use models
2087

FIG. 20H

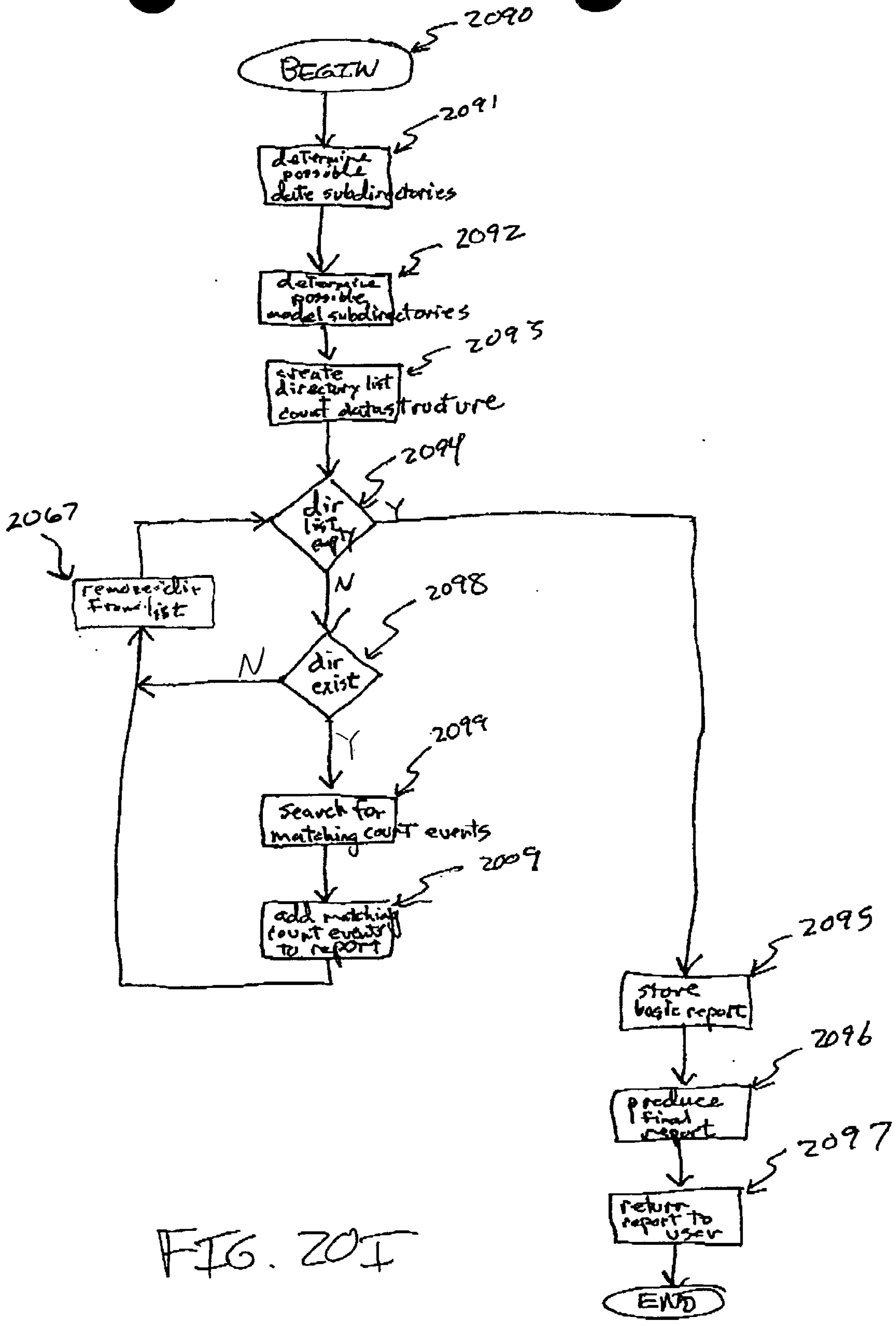

2090
BEGIN
2091
determine possible date subdirectories
2092
determine possible model subdirectories
2093
create directory list count data structure
2094
dir list empty
Y
N
2067
remove dir from list
2098
dir exist
N
Y
2099
search for matching count events
2009
add matching count events to report
2095
store basic report
2096
produce final report
2097
return report to user
END

FIG. 20I

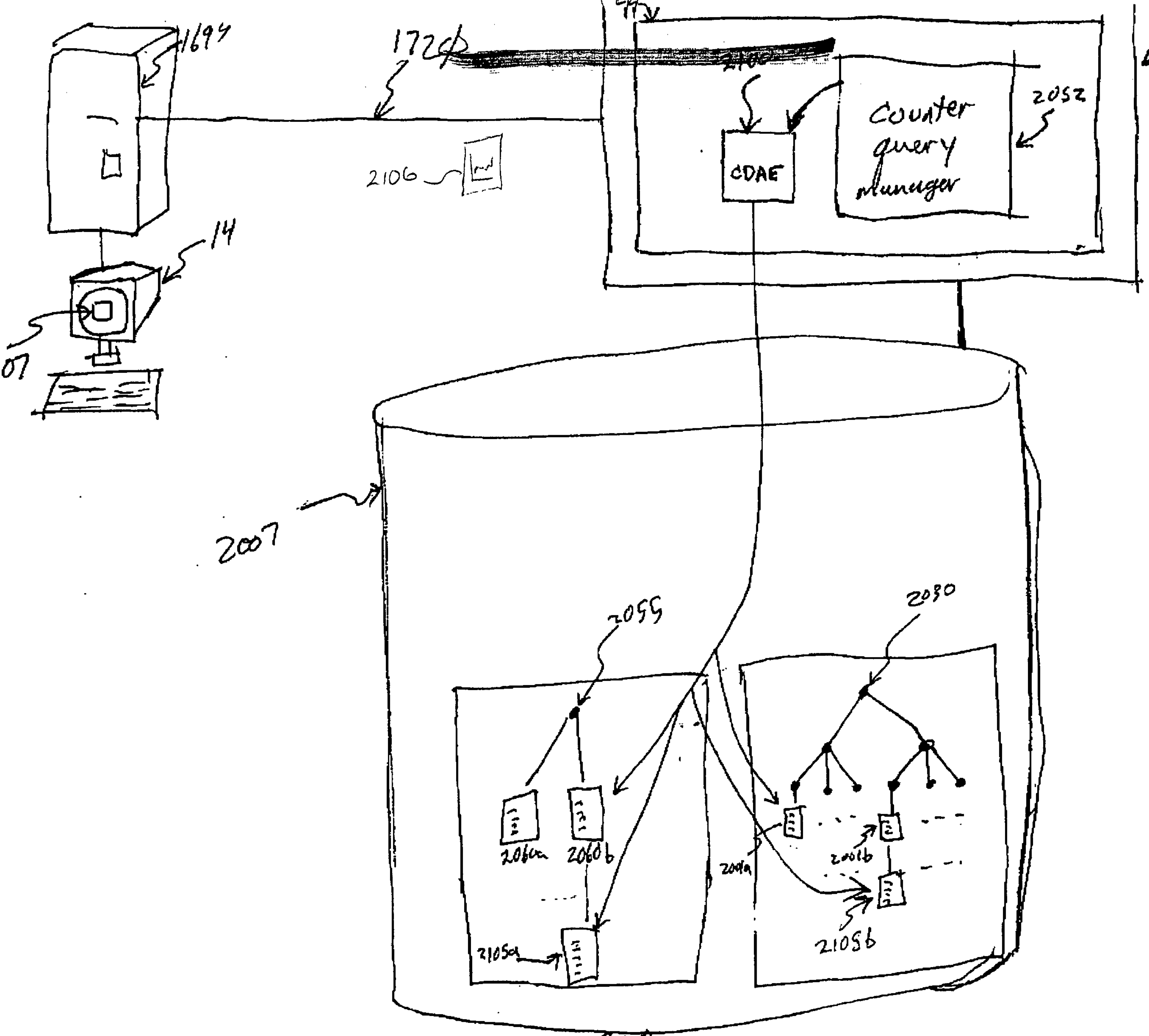

1694
172d
2106
14
2107
44
2100
CDAE
Counter query manager
2052
1698
2007
2055
2060a
2060b
2105a
2030
2001a
2001b
2105b

FIG. 21A

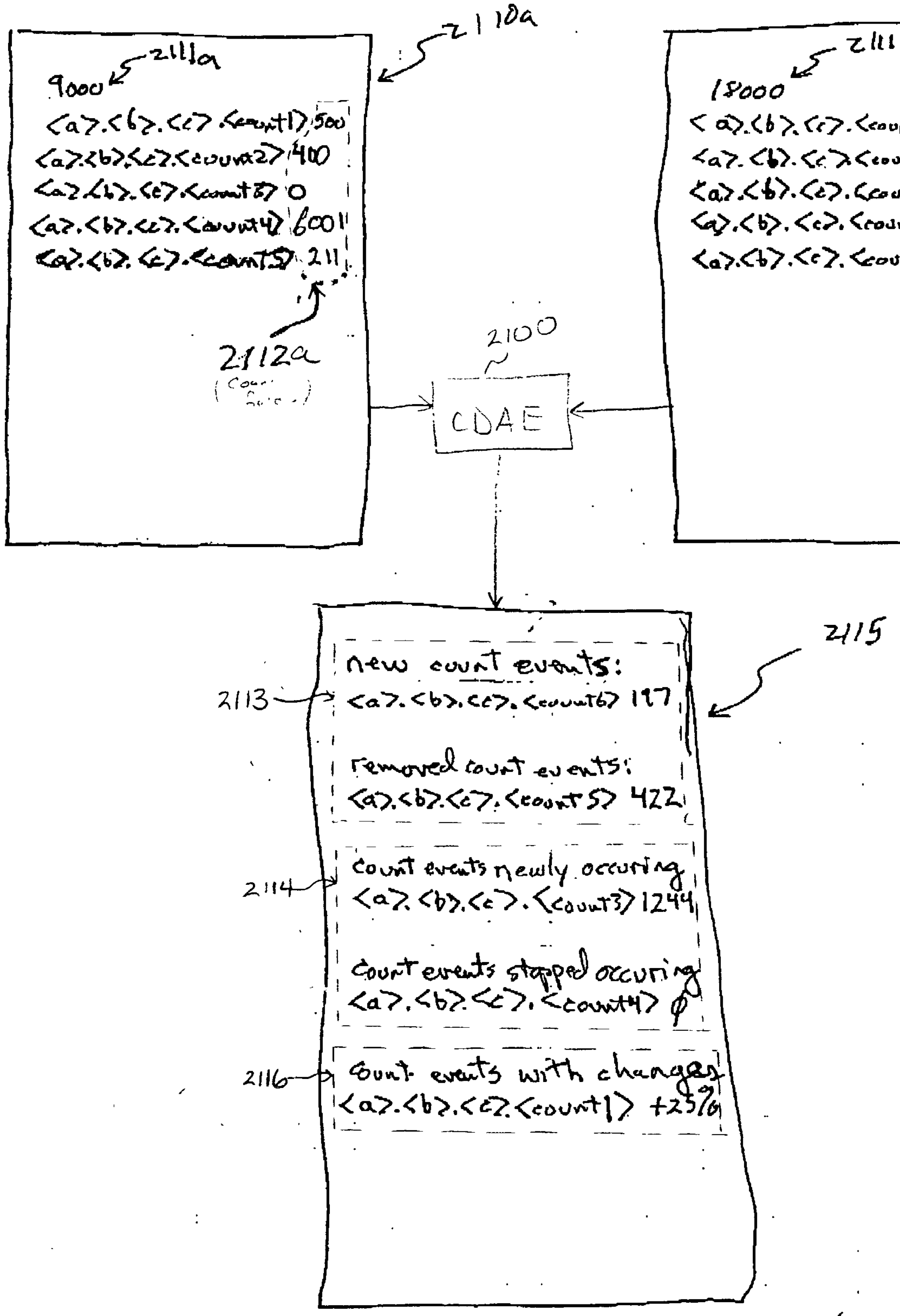

2110a
2111a
9000
<a>.<b>.<c>.<count1> 500
<a>.<b>.<c>.<count2> 400
<a>.<b>.<c>.<count3> 0
<a>.<b>.<c>.<count4> 6001
<a>.<b>.<c>.<count5> 211
2112a
(Count Series)
2110b
2111b
18000
<a>.<b>.<c>.<count1> 1250
<a>.<b>.<c>.<count2> 812
<a>.<b>.<c>.<count3> 1244
<a>.<b>.<c>.<count4> 0
<a>.<b>.<c>.<count6> 197
2112b
2100
CDAE
2115
new count events:
2113
<a>.<b>.<c>.<count6> 197
removed count events:
<a>.<b>.<c>.<count5> 422
count events newly occuring
2114
<a>.<b>.<c>.<count3> 1244
count events stopped occuring
<a>.<b>.<c>.<count4> ϕ
2116
count events with changes
<a>.<b>.<c>.<count1> +25%

FIG. 21B

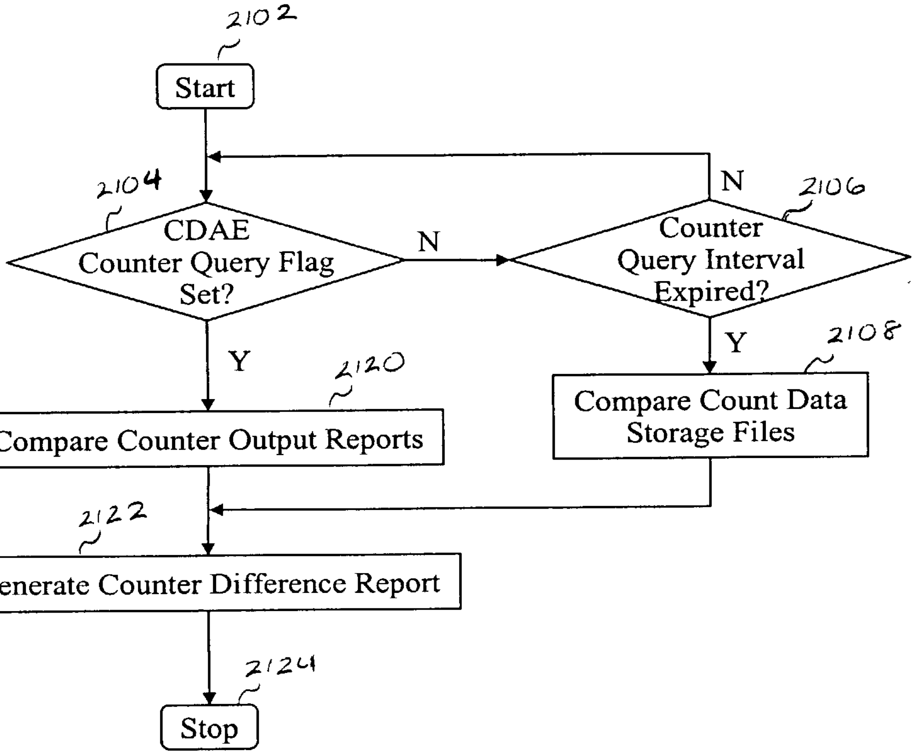

2102
Start
2104
CDAE Counter Query Flag Set?
N
2106
Counter Query Interval Expired?
N
Y
2108
Compare Count Data Storage Files
Y
2120
Compare Counter Output Reports
2122
Generate Counter Difference Report
2124
Stop

FIG. 21C

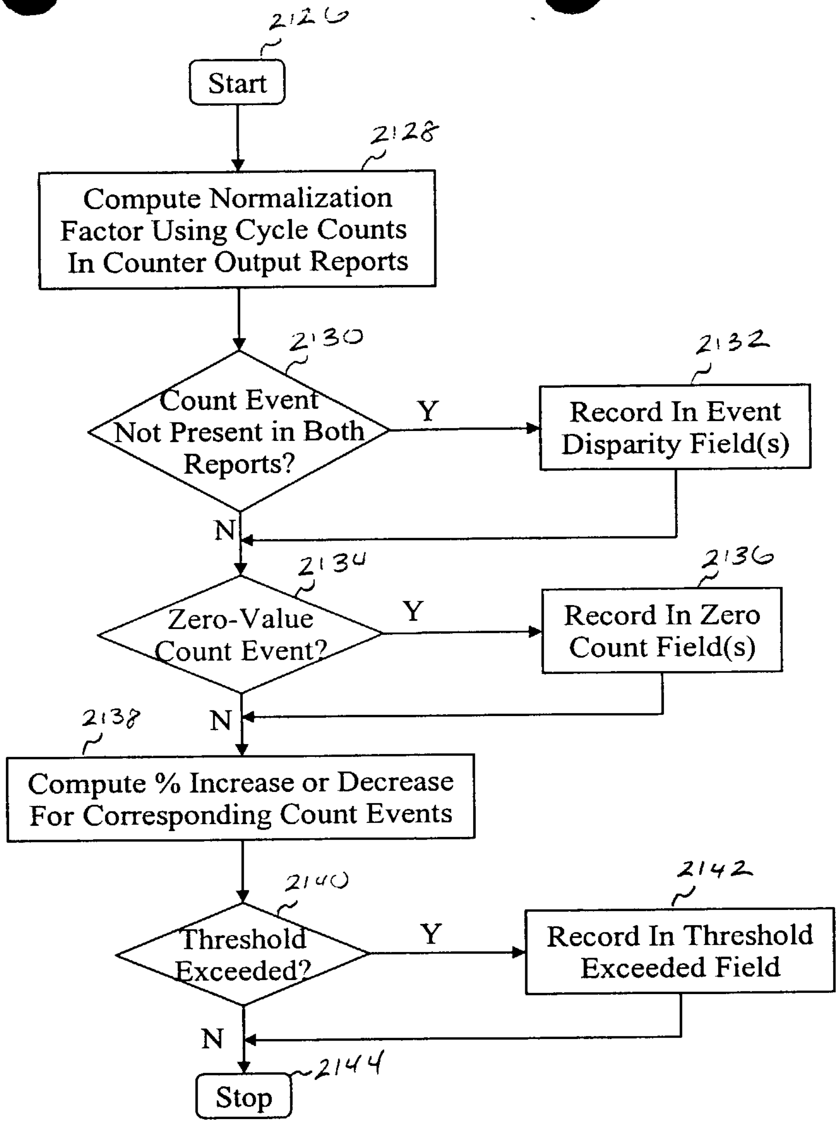

2126
Start
2128
Compute Normalization Factor Using Cycle Counts In Counter Output Reports
2130
Count Event Not Present in Both Reports?
Y
2132
Record In Event Disparity Field(s)
N
2134
Zero-Value Count Event?
Y
2136
Record In Zero Count Field(s)
N
2138
Compute % Increase or Decrease For Corresponding Count Events
2140
Threshold Exceeded?
Y
2142
Record In Threshold Exceeded Field
N
2144
Stop

FIG. 21D

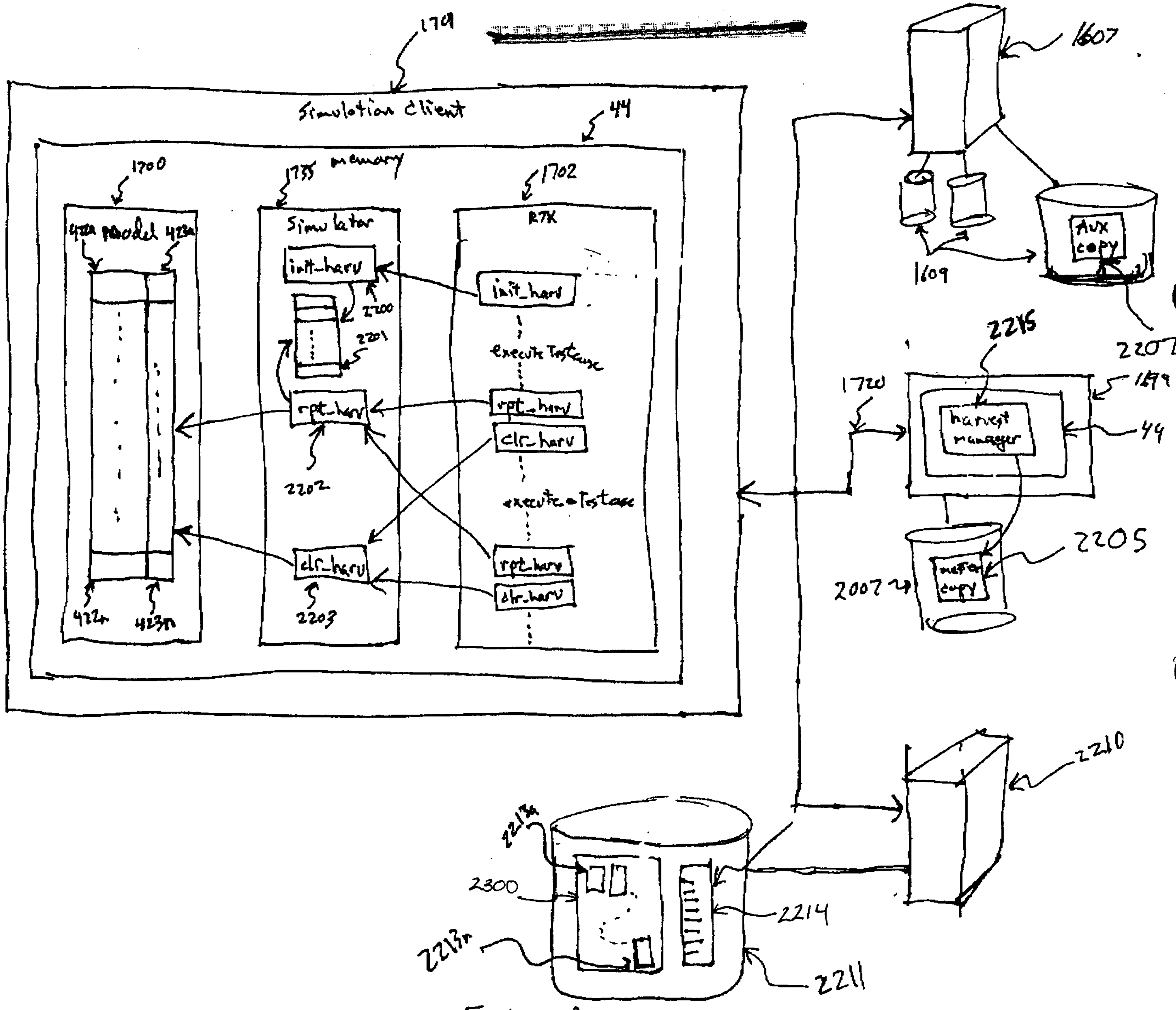

179
Simulation client
44
memory
1700
model
422a
423a
422n
423n
1735
Simulator
init_harv
2200
2201
rpt_harv
2202
clr_harv
2203
1702
RTX
init_harv
execute Testcase
rpt_harv
clr_harv
execute Testcase
rpt_harv
clr_harv
1607
1609
Aux copy
2207a
2245
harvest manager
1799
44
1720
master copy
2205
2007
2210
2213a
2300
2213n
2214
2211

FIG. 22A

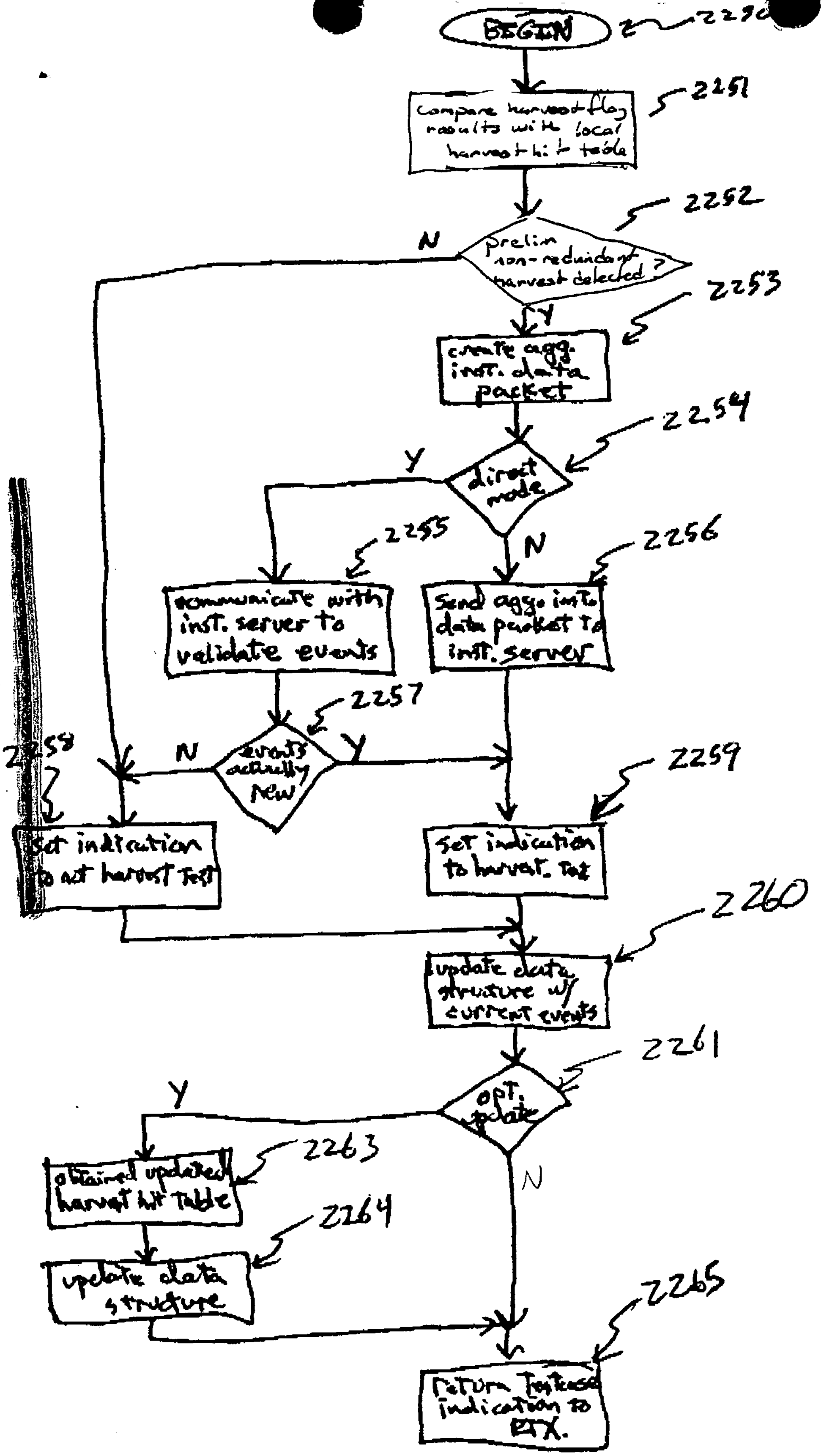

BEGIN
2250
compare harvest flag results with local harvest hit table
2251
prelim non-redundant harvest detected ?
2252
N
Y
create agg. inst. data packet
2253
direct mode
2254
Y
N
communicate with inst. server to validate events
2255
Send agg. info data packet to inst. server
2256
events actually new
2257
N
Y
2258
set indication to not harvest test
set indication to harvest. test
2259
update data structure w/ current events
2260
opt. update
2261
Y
N
obtained updated harvest hit Table
2263
update data structure
2264
return Testcase indication to RTX.
2265

FIG. 22B

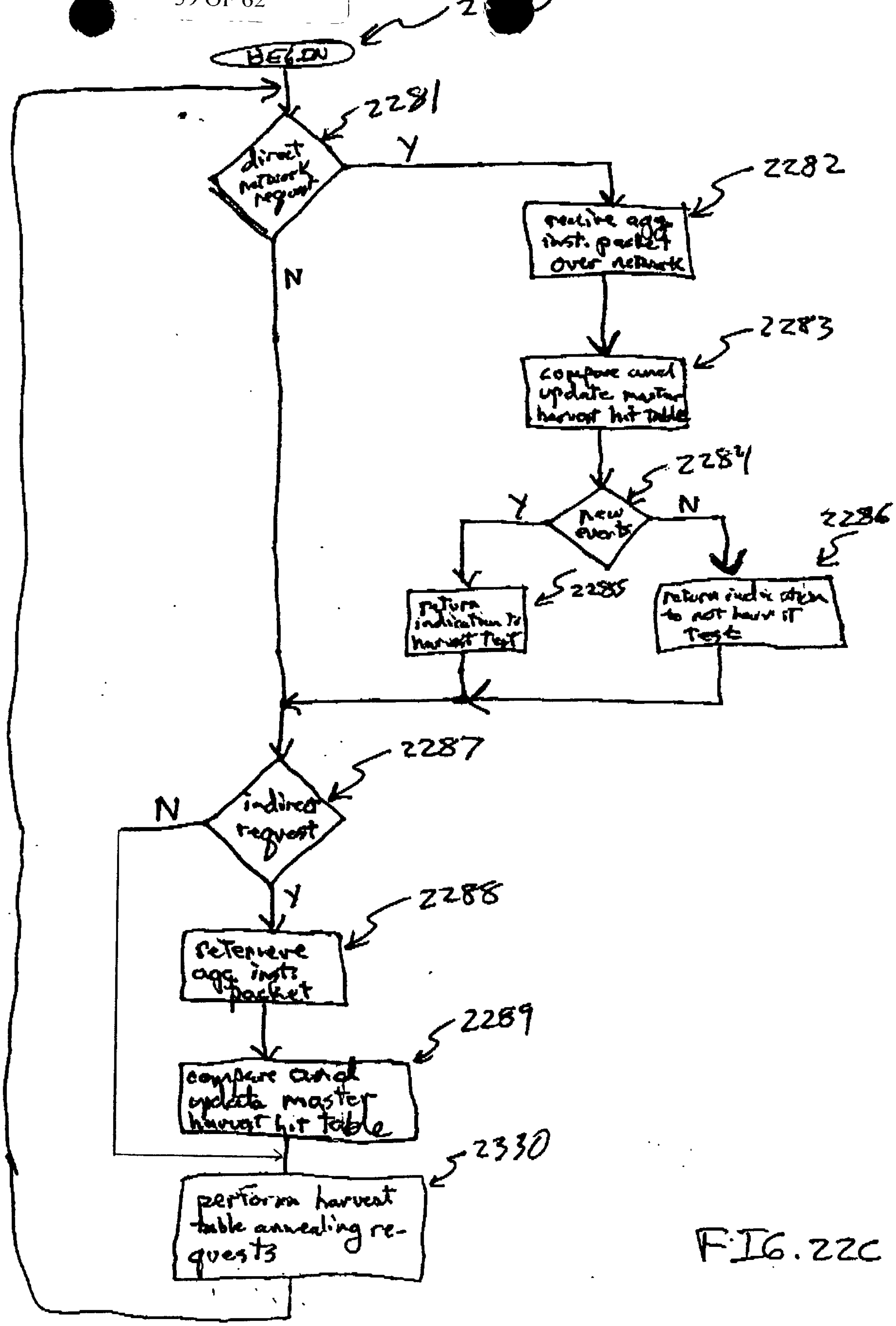

22
BEGIN
2281
direct network request
Y
N
2282
receive agg. inst. packet over network
2283
compare and update master harvest hit table
2284
new events
Y
N
2285
return indication to harvest Test
2286
return indication to not harvest Test
2287
indirect request
N
Y
2288
retrieve agg. insts packet
2289
compare and update master harvest hit table
2330
perform harvest table annealing requests

FIG. 22C

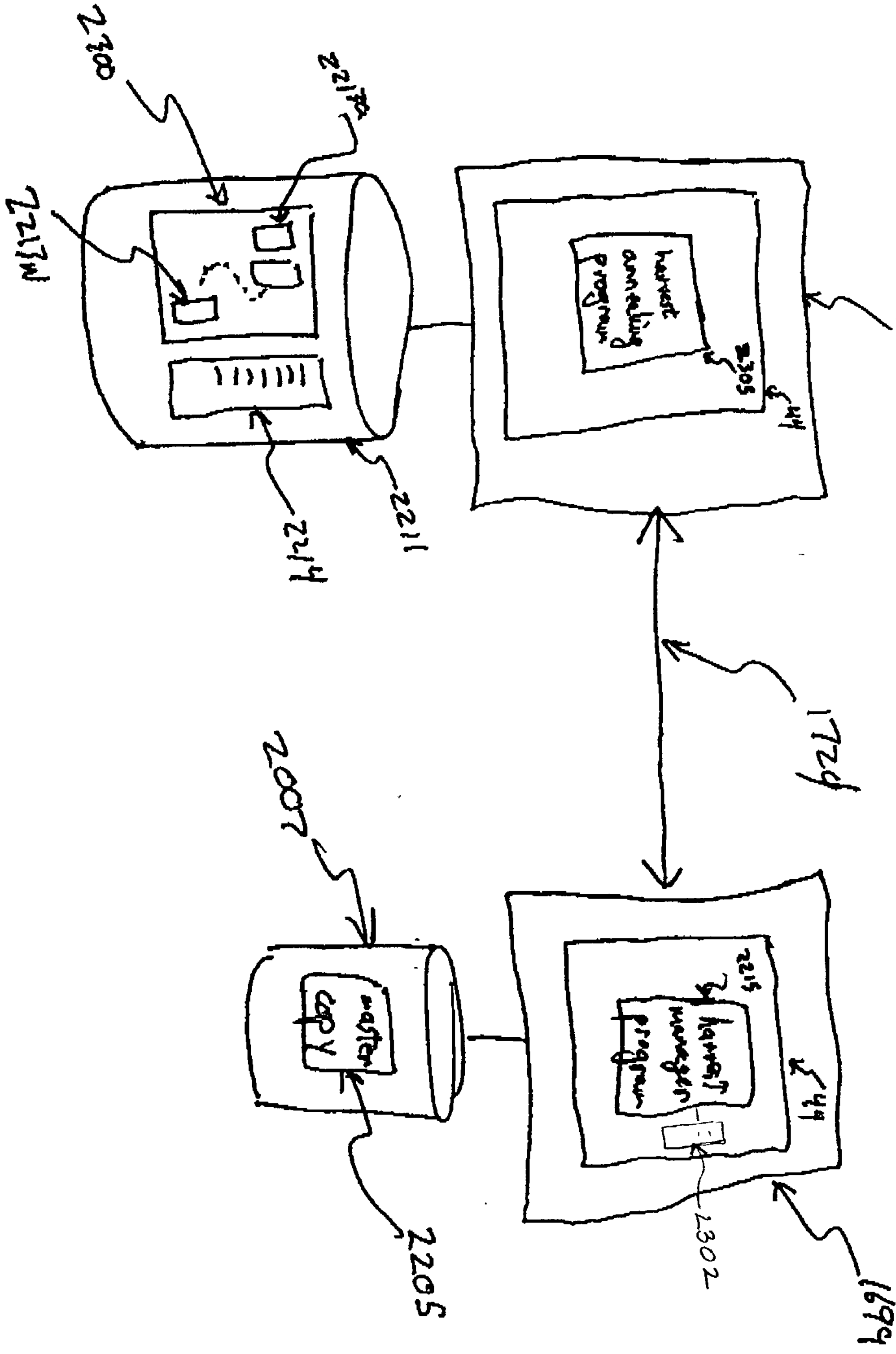
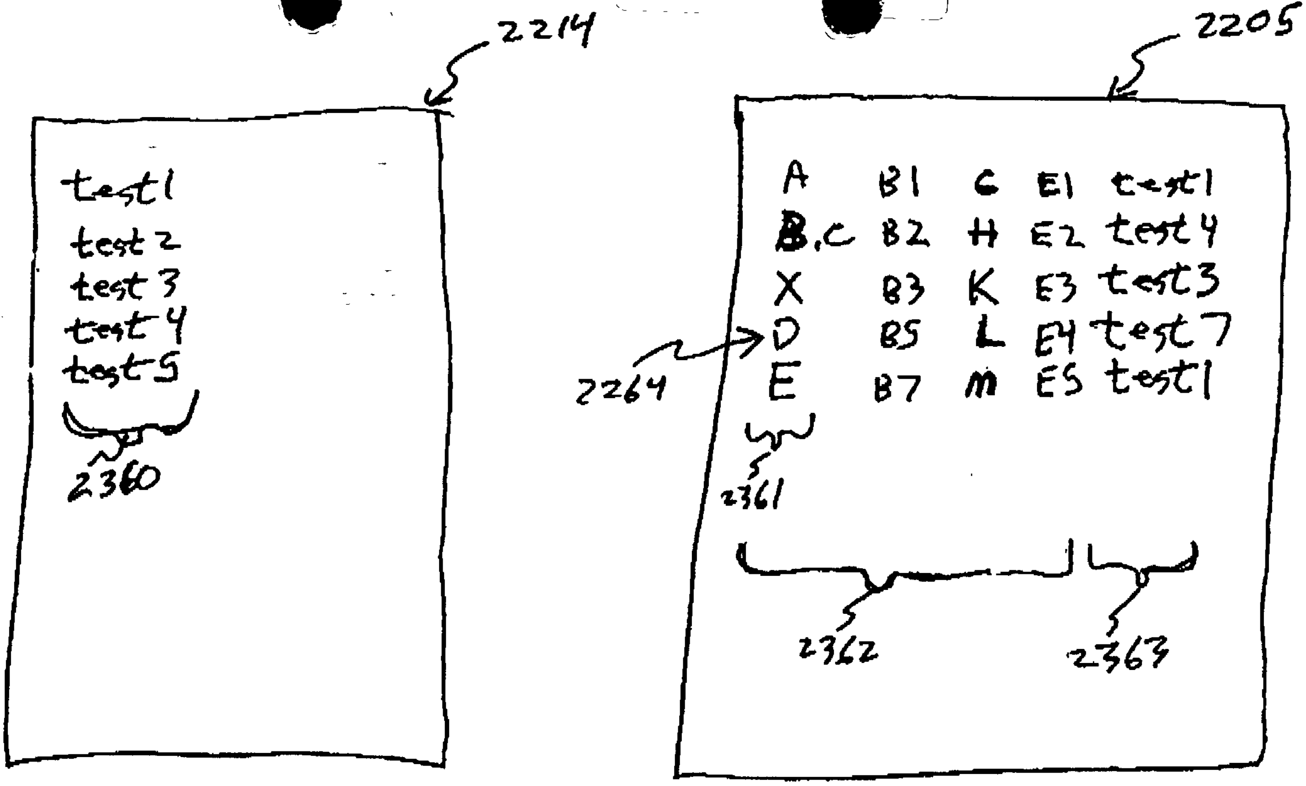

2300
2217A
2217N
2211
2214
harvest
controlling
program
2305
44
2210
1724
2007
master
copy
2205
2215
harvest
manager
program
44
2302
1694

FIG. 23A



test1
test2
test3
test4
test5

| | | | | |
|---|---|---|---|---|
| A | B1 | C | E1 | test1 |
| B,C | B2 | H | E2 | test4 |
| X | B3 | K | E3 | test3 |
| D | B5 | L | E4 | test7 |
| E | B7 | M | E5 | test1 |

FIG. 23B

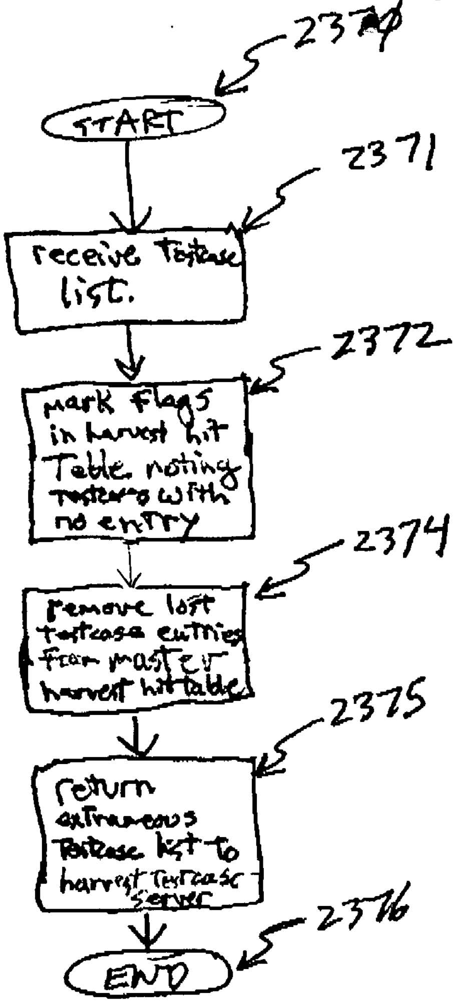

2370
START
2371
receive Testcase list.
2372
mark flags in harvest hit Table. noting Testcases with no entry
2374
remove lost testcase entries from master harvest hit table
2375
return extraneous Testcase list to harvest Testcase server
2376
END

FIG. 23Z

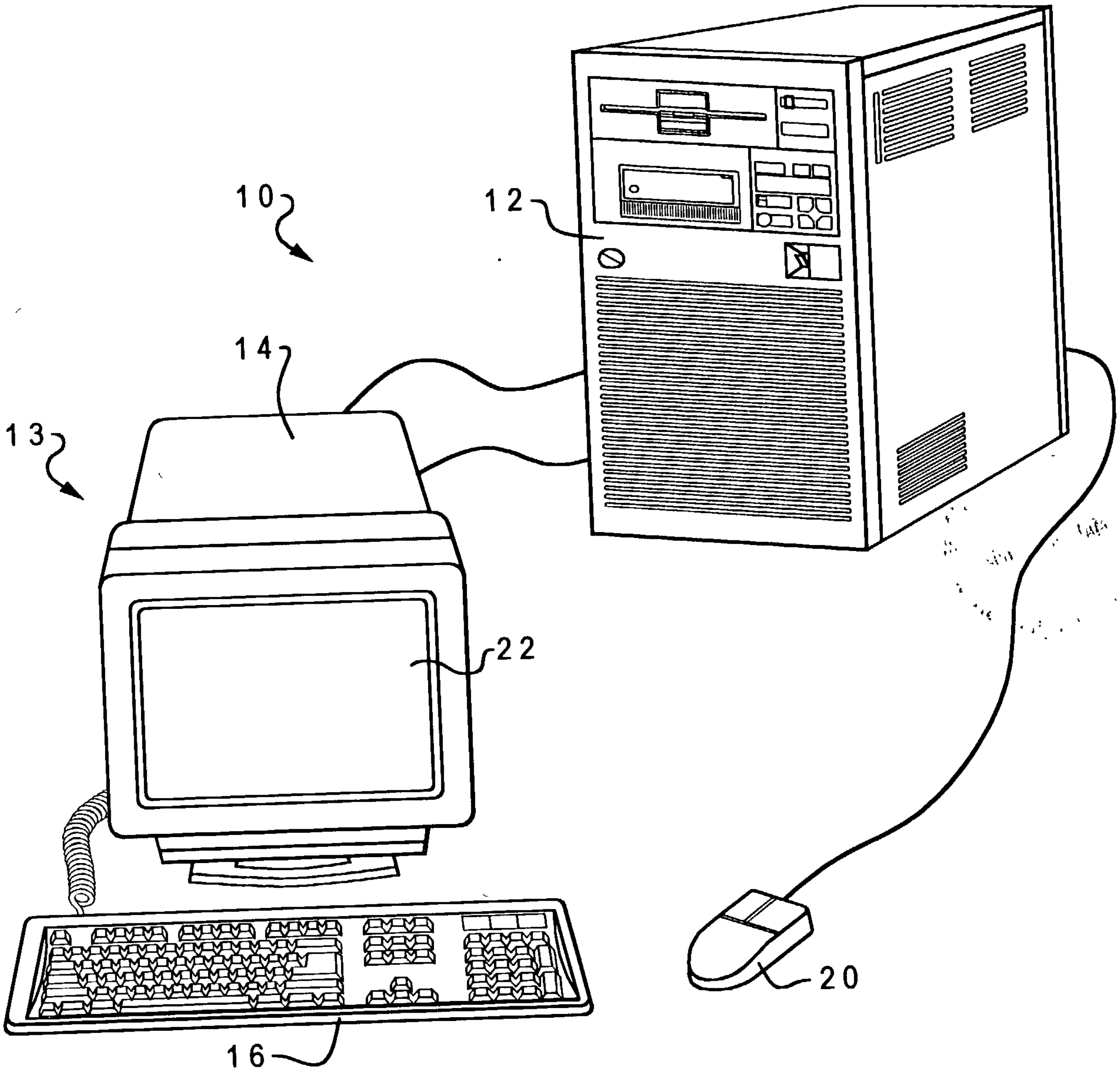


*Fig. 1*

*Prior Art*

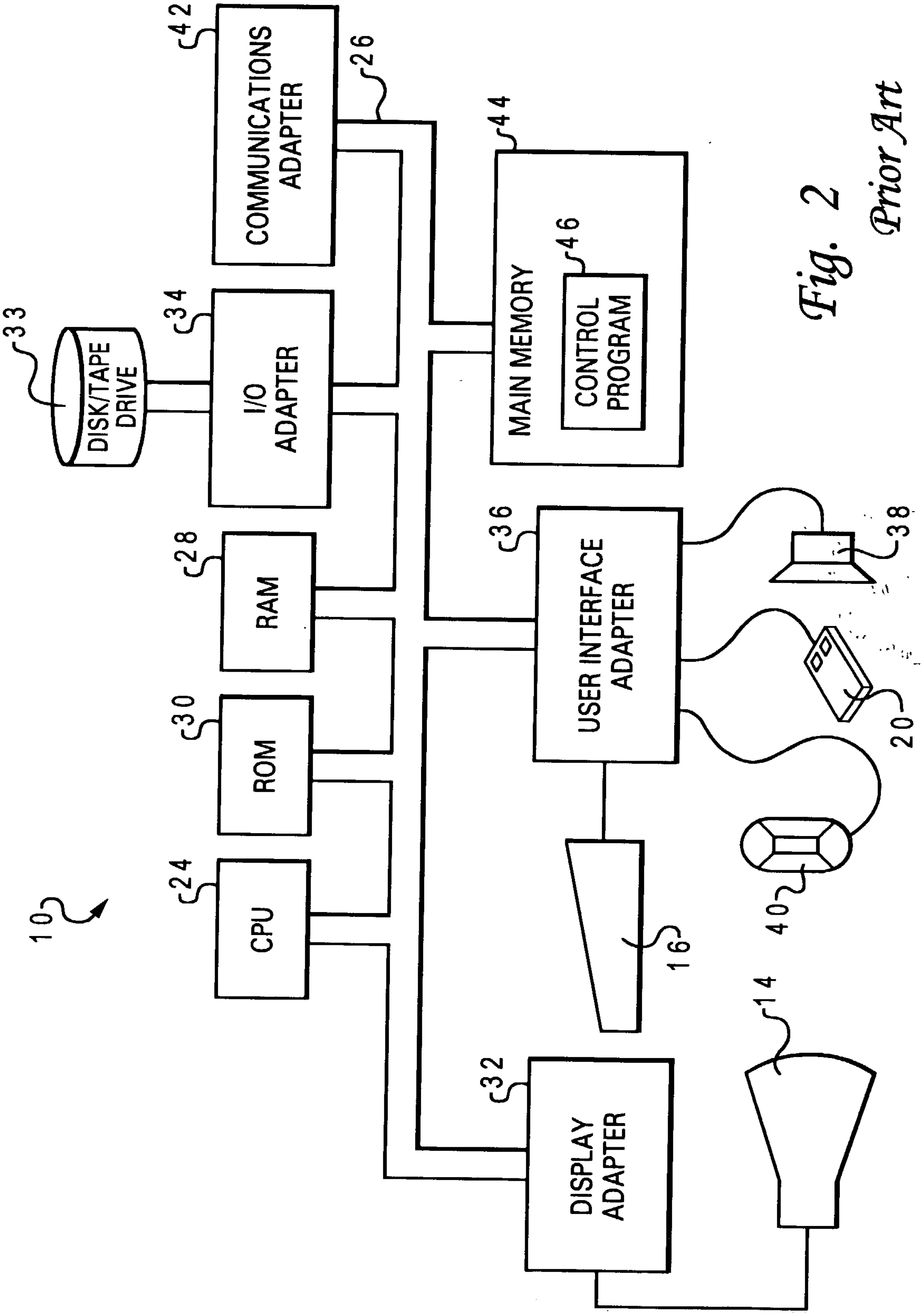


*Fig. 2*
*Prior Art*

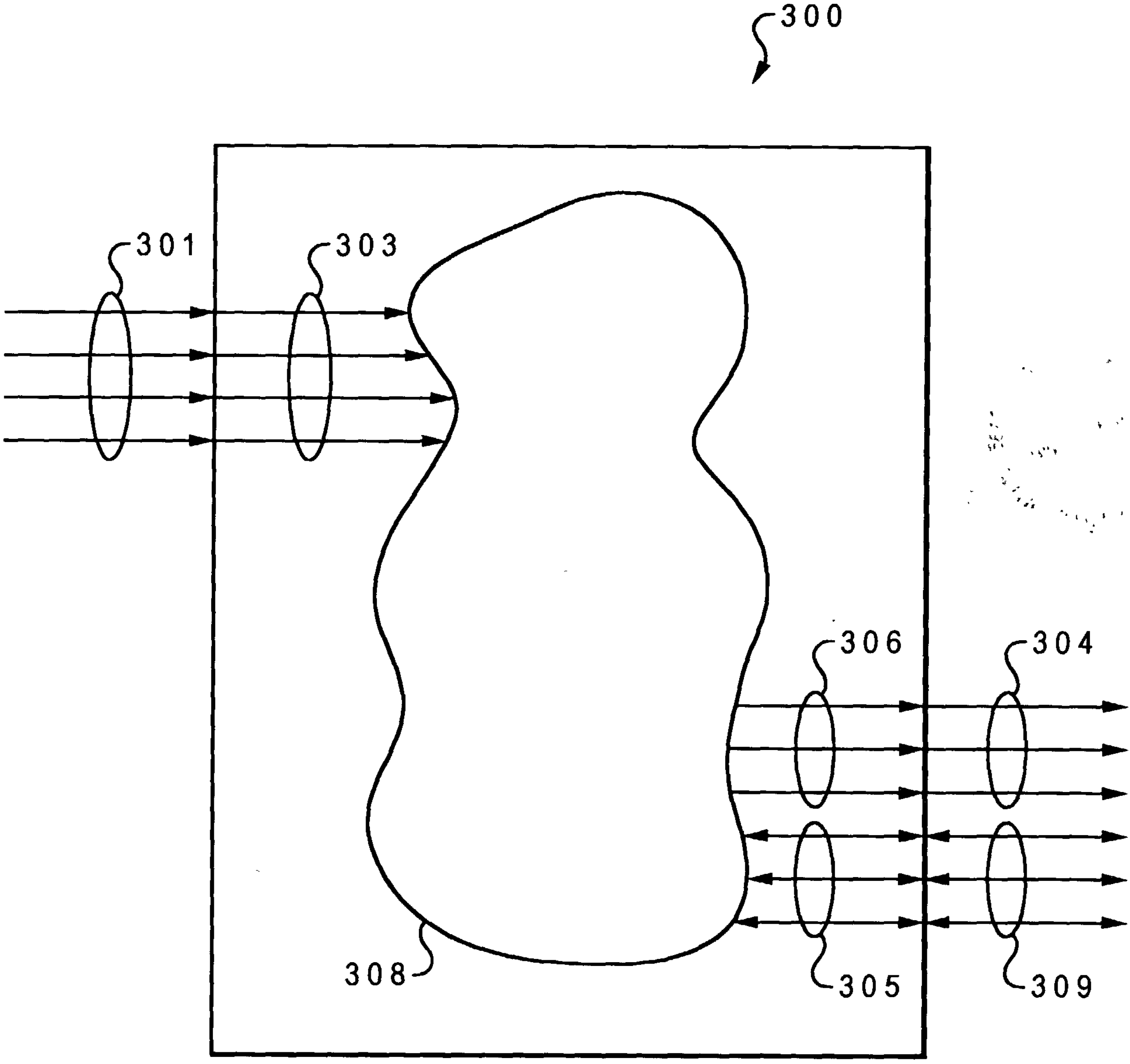


*Fig. 3A*

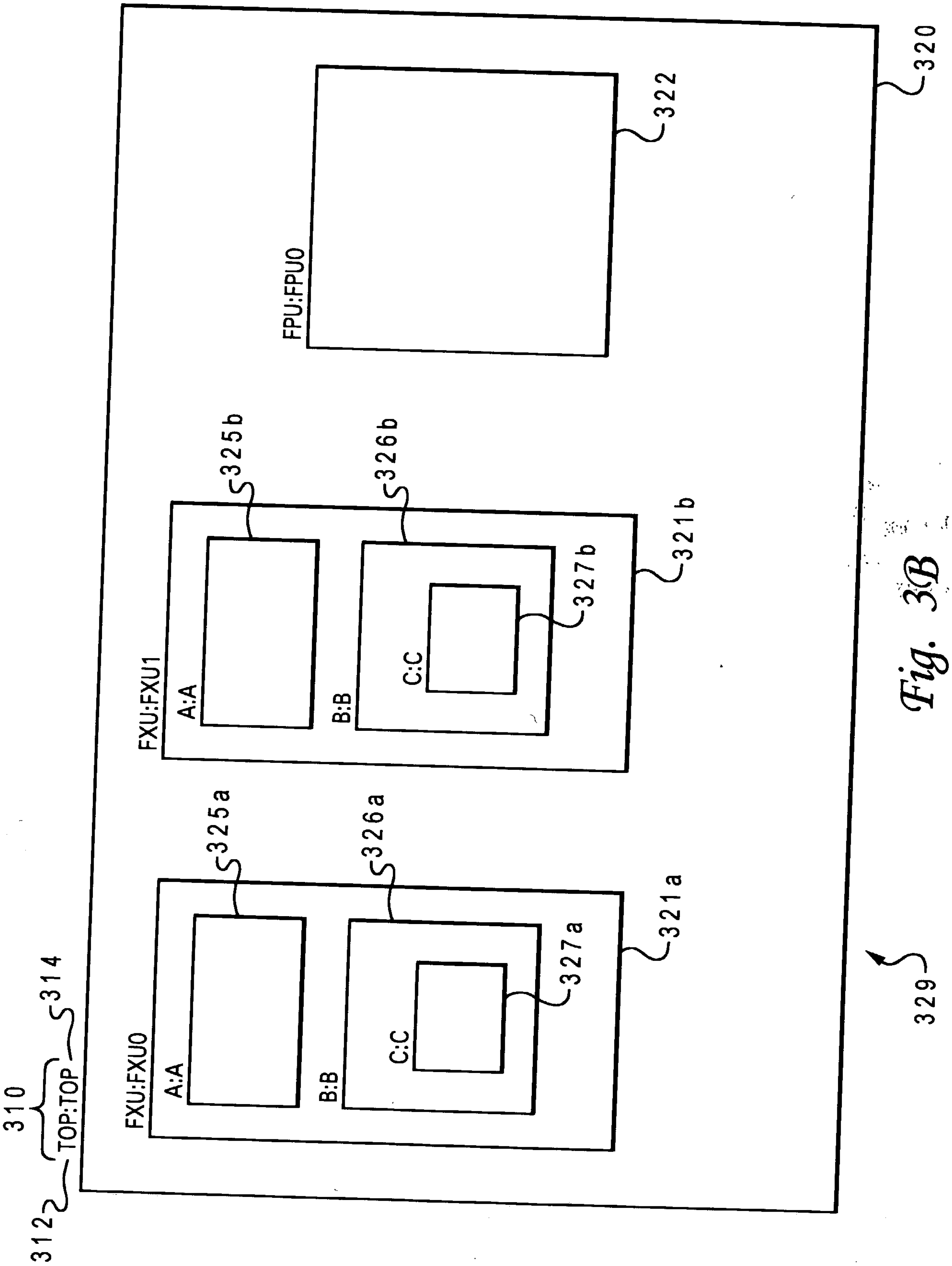

310
TOP:TOP
312
314
FXU:FXU0
A:A
B:B
C:C
325a
326a
327a
321a
FXU:FXU1
A:A
B:B
C:C
325b
326b
327b
321b
FPU:FPU0
322
320
329

Fig. 3B

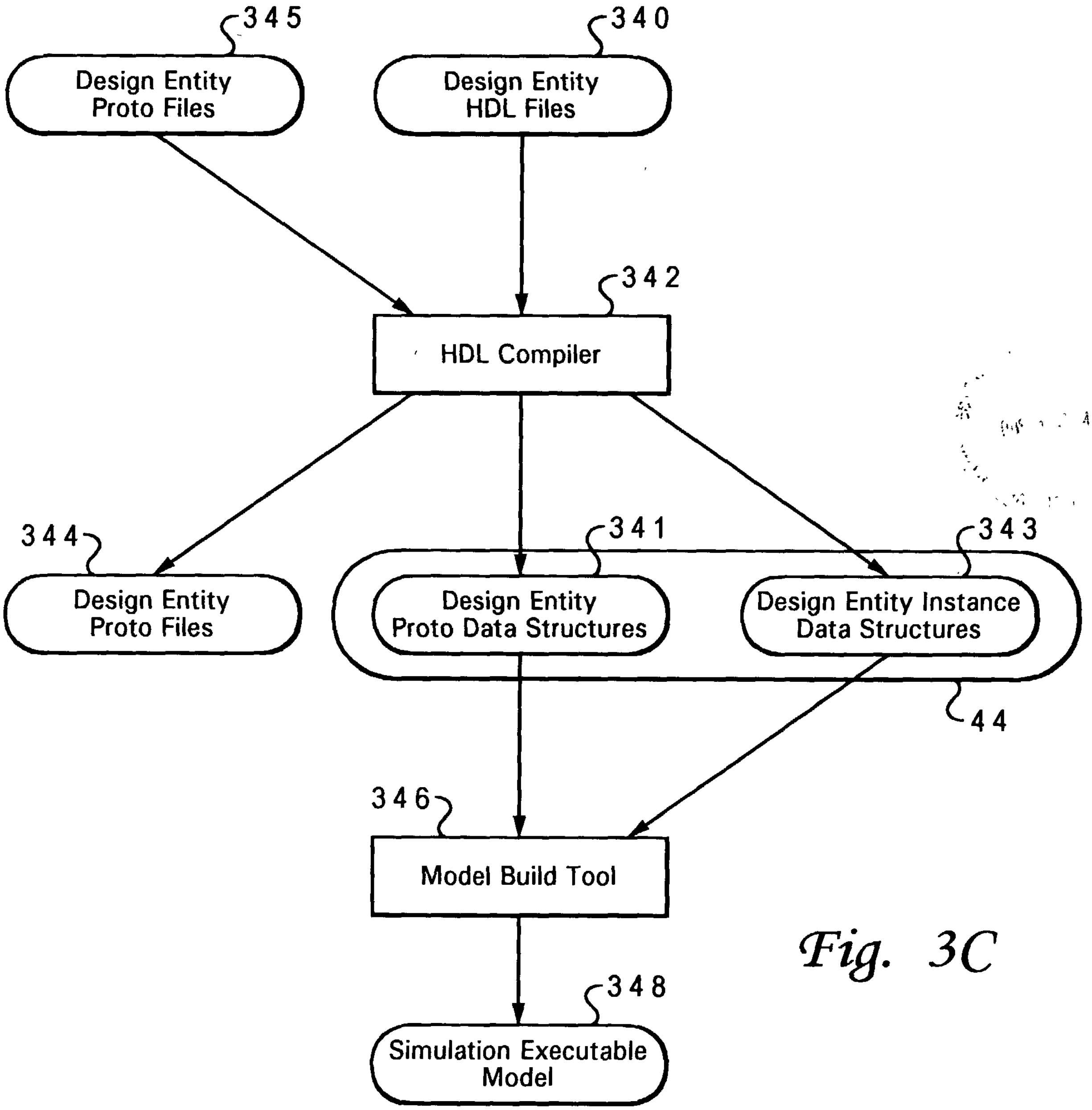

345
Design Entity Proto Files
340
Design Entity HDL Files
342
HDL Compiler
344
Design Entity Proto Files
341
Design Entity Proto Data Structures
343
Design Entity Instance Data Structures
44
346
Model Build Tool
348
Simulation Executable Model

*Fig. 3C*

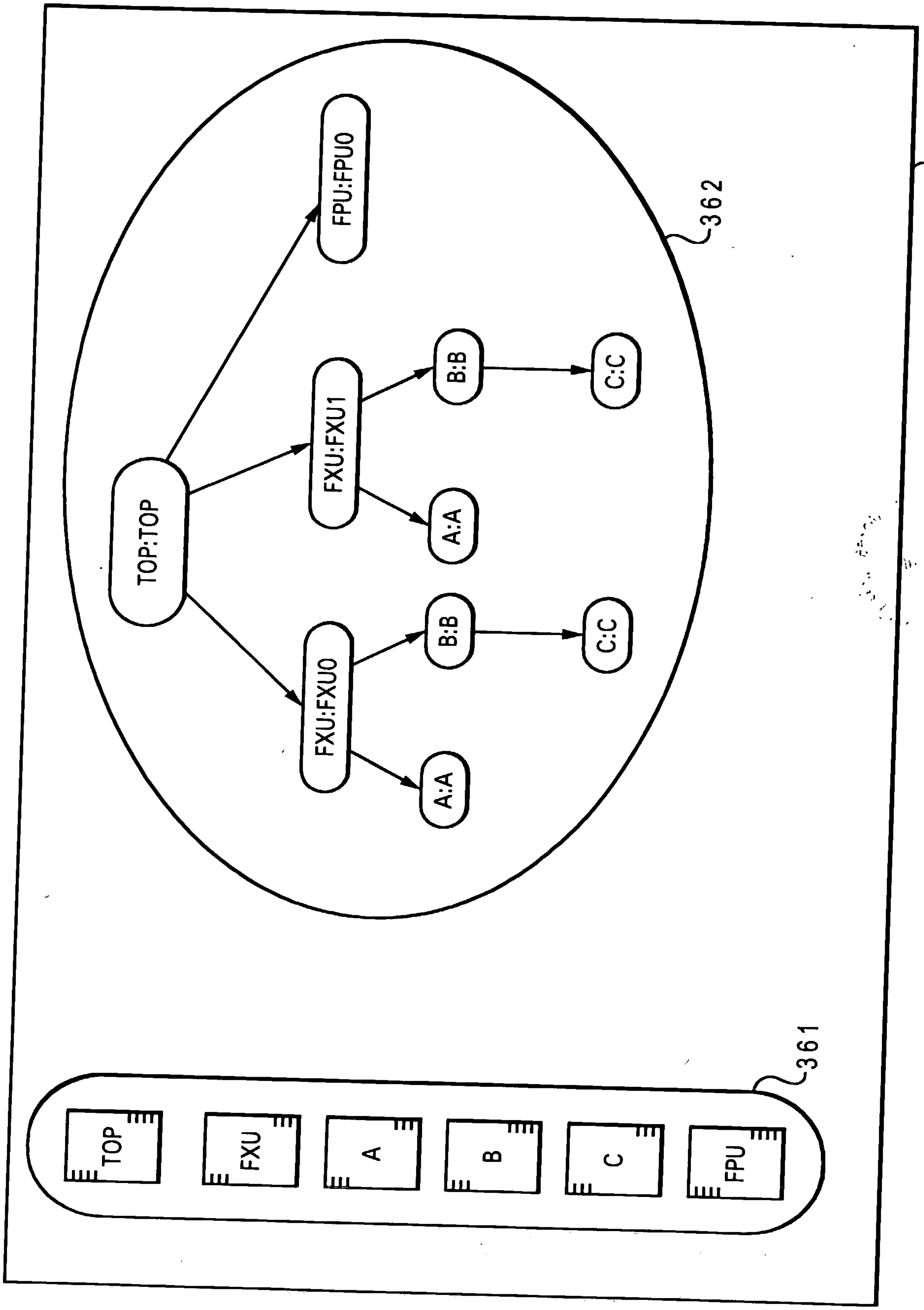

TOP:TOP
FXU:FXU0
FXU:FXU1
FPU:FPU0
A:A
B:B
C:C
A:A
B:B
C:C
362
44
TOP
FXU
A
B
C
FPU
361

*Fig. 3D*

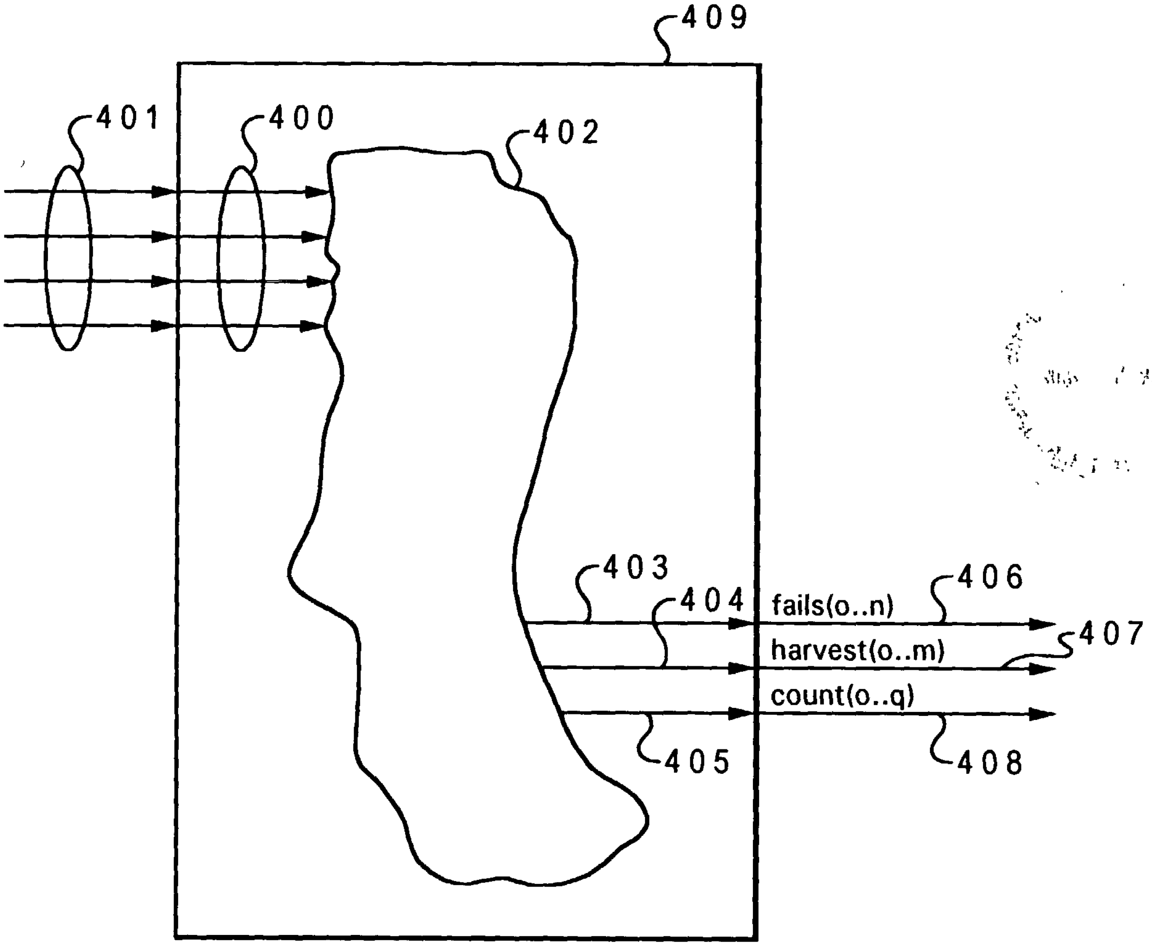

409
401
400
402
403
404
fails(o..n)
406
407
harvest(o..m)
count(o..q)
405
408

*Fig. 4A*

AUS920000651US1

Gabele, et al.
n-Redundant Collection Of Harvest Events
Within A Batch Simulation Farm Network

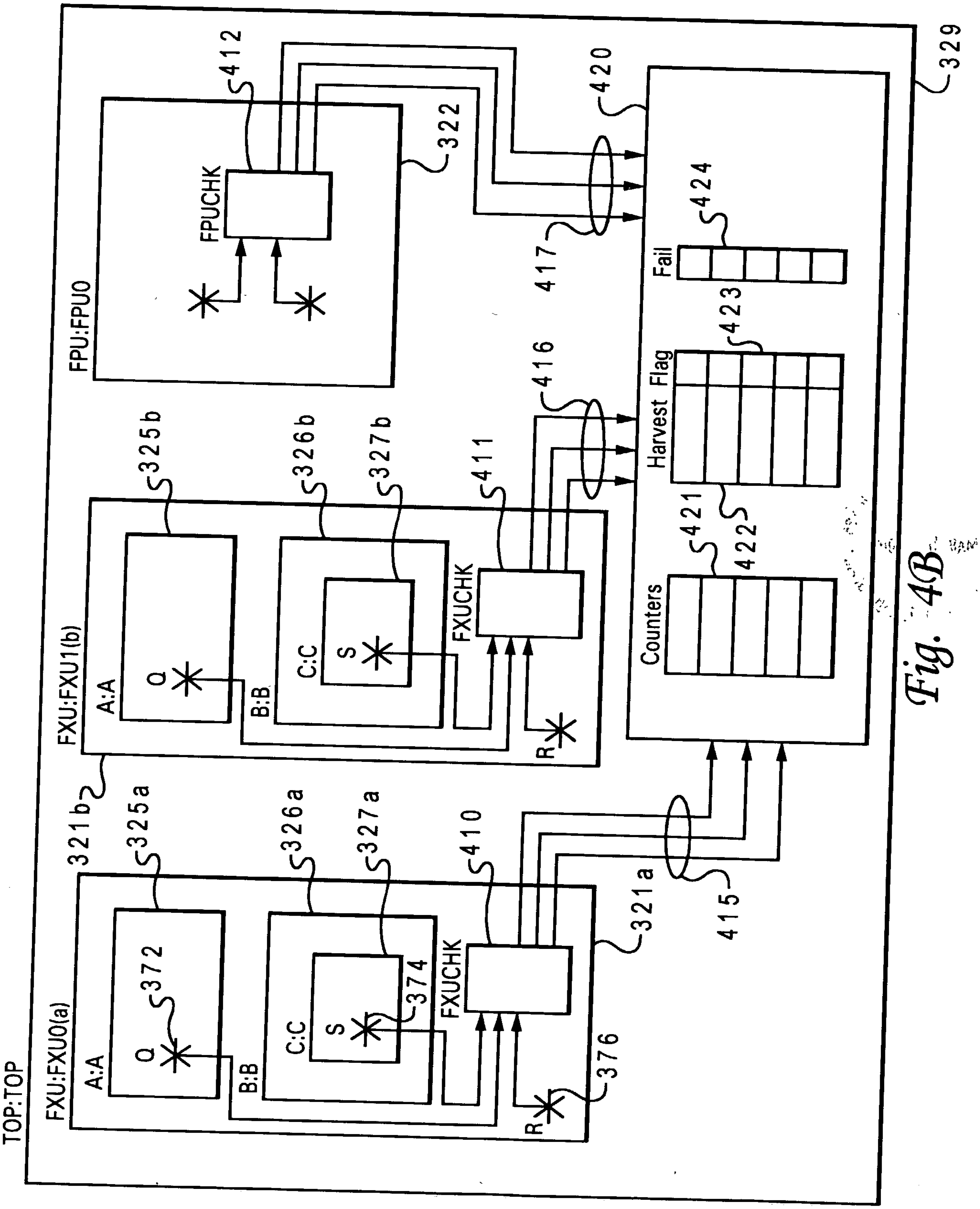
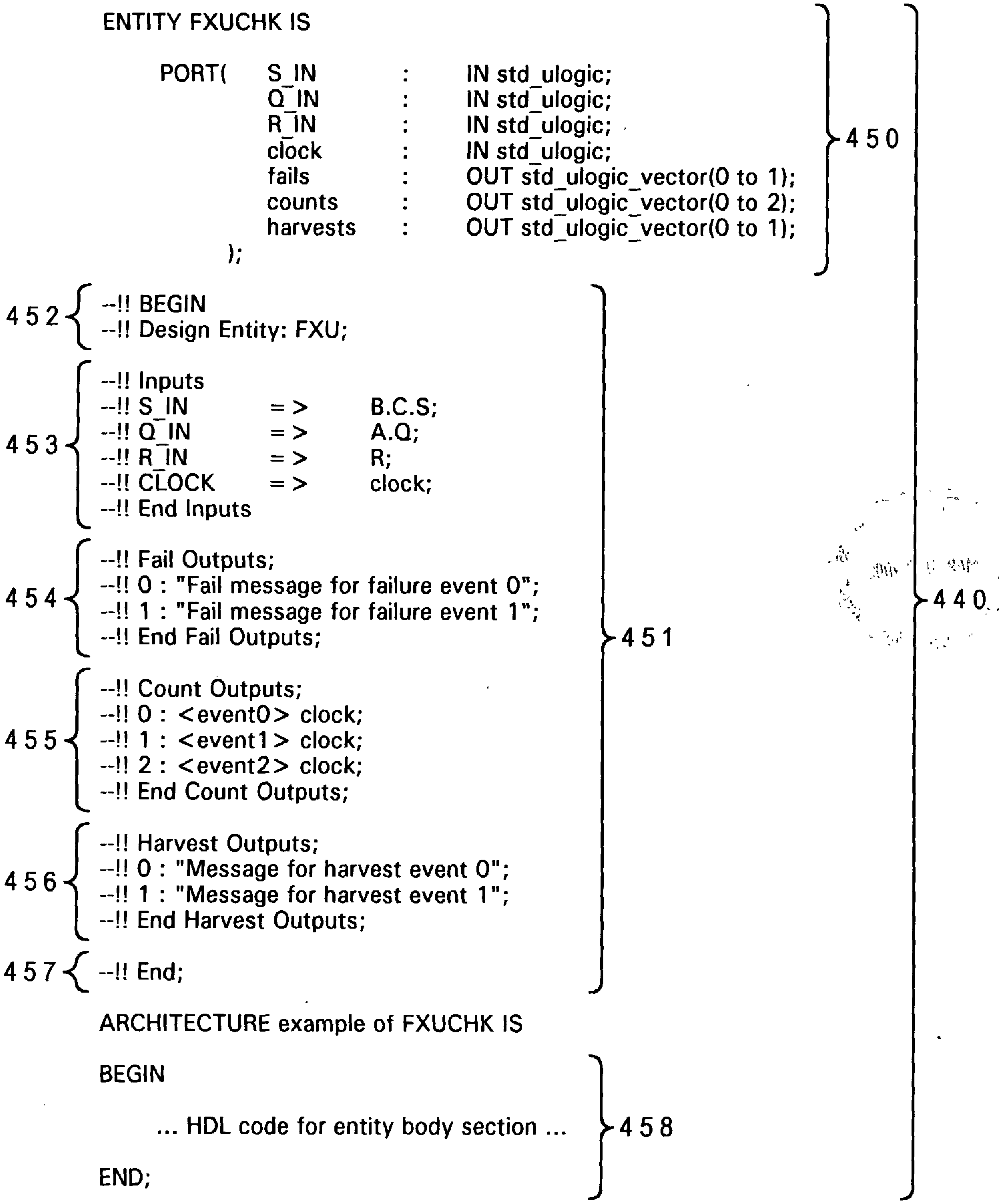


*Fig. 4B*



```
ENTITY FXUCHK IS

    PORT(   S_IN        :       IN std_ulogic;
            Q_IN        :       IN std_ulogic;
            R_IN        :       IN std_ulogic;
            clock       :       IN std_ulogic;
            fails       :       OUT std_ulogic_vector(0 to 1);
            counts      :       OUT std_ulogic_vector(0 to 2);
            harvests    :       OUT std_ulogic_vector(0 to 1);
        );

--!! BEGIN
--!! Design Entity: FXU;

--!! Inputs
--!! S_IN       =>      B.C.S;
--!! Q_IN       =>      A.Q;
--!! R_IN       =>      R;
--!! CLOCK      =>      clock;
--!! End Inputs

--!! Fail Outputs;
--!! 0 : "Fail message for failure event 0";
--!! 1 : "Fail message for failure event 1";
--!! End Fail Outputs;

--!! Count Outputs;
--!! 0 : <event0> clock;
--!! 1 : <event1> clock;
--!! 2 : <event2> clock;
--!! End Count Outputs;

--!! Harvest Outputs;
--!! 0 : "Message for harvest event 0";
--!! 1 : "Message for harvest event 1";
--!! End Harvest Outputs;

--!! End;

ARCHITECTURE example of FXUCHK IS

BEGIN

      ... HDL code for entity body section ...

END;
```

*Fig. 4C*

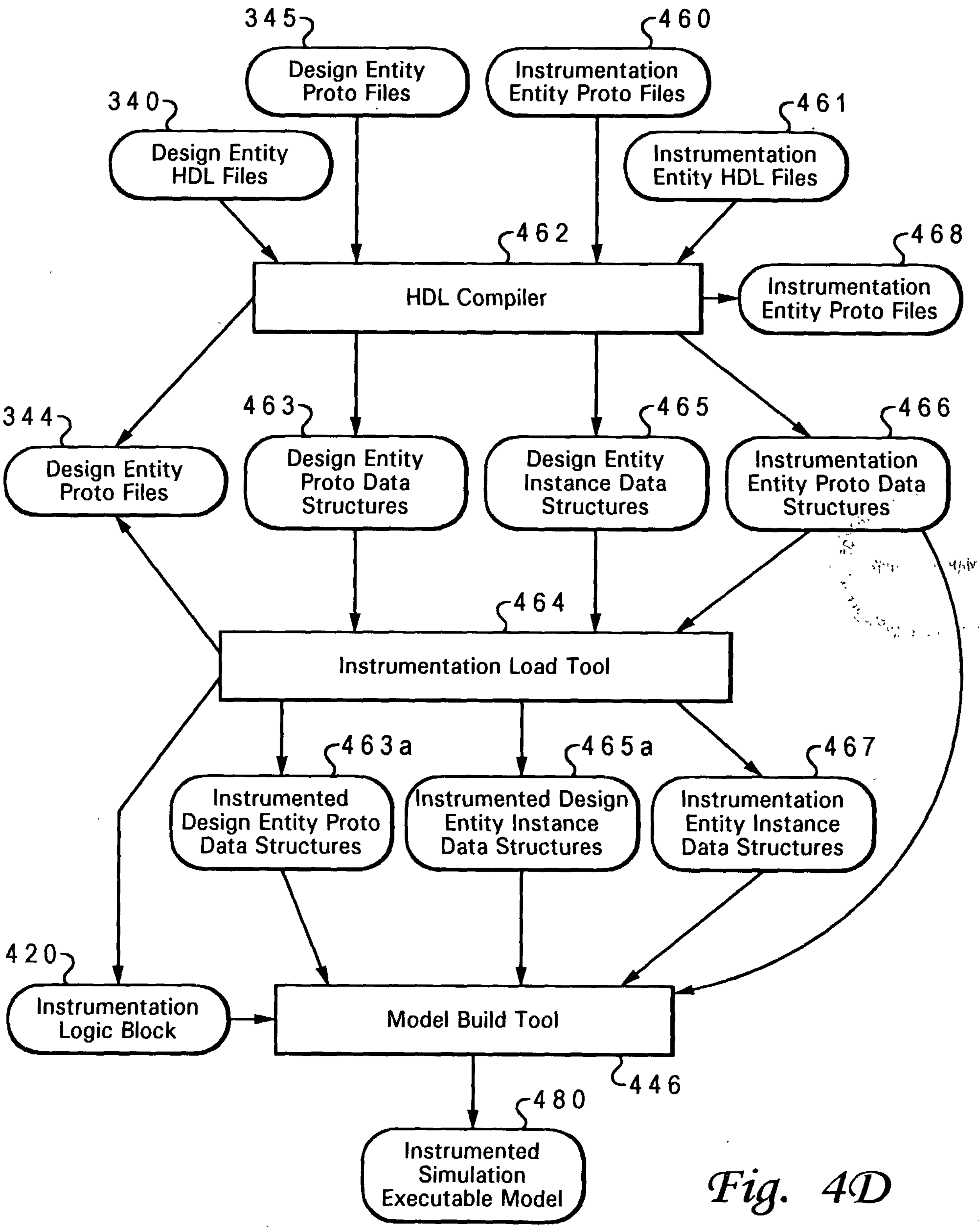

345
Design Entity Proto Files
460
Instrumentation Entity Proto Files
340
Design Entity HDL Files
461
Instrumentation Entity HDL Files
462
HDL Compiler
468
Instrumentation Entity Proto Files
344
Design Entity Proto Files
463
Design Entity Proto Data Structures
465
Design Entity Instance Data Structures
466
Instrumentation Entity Proto Data Structures
464
Instrumentation Load Tool
463a
Instrumented Design Entity Proto Data Structures
465a
Instrumented Design Entity Instance Data Structures
467
Instrumentation Entity Instance Data Structures
420
Instrumentation Logic Block
Model Build Tool
446
480
Instrumented Simulation Executable Model

*Fig. 4D*

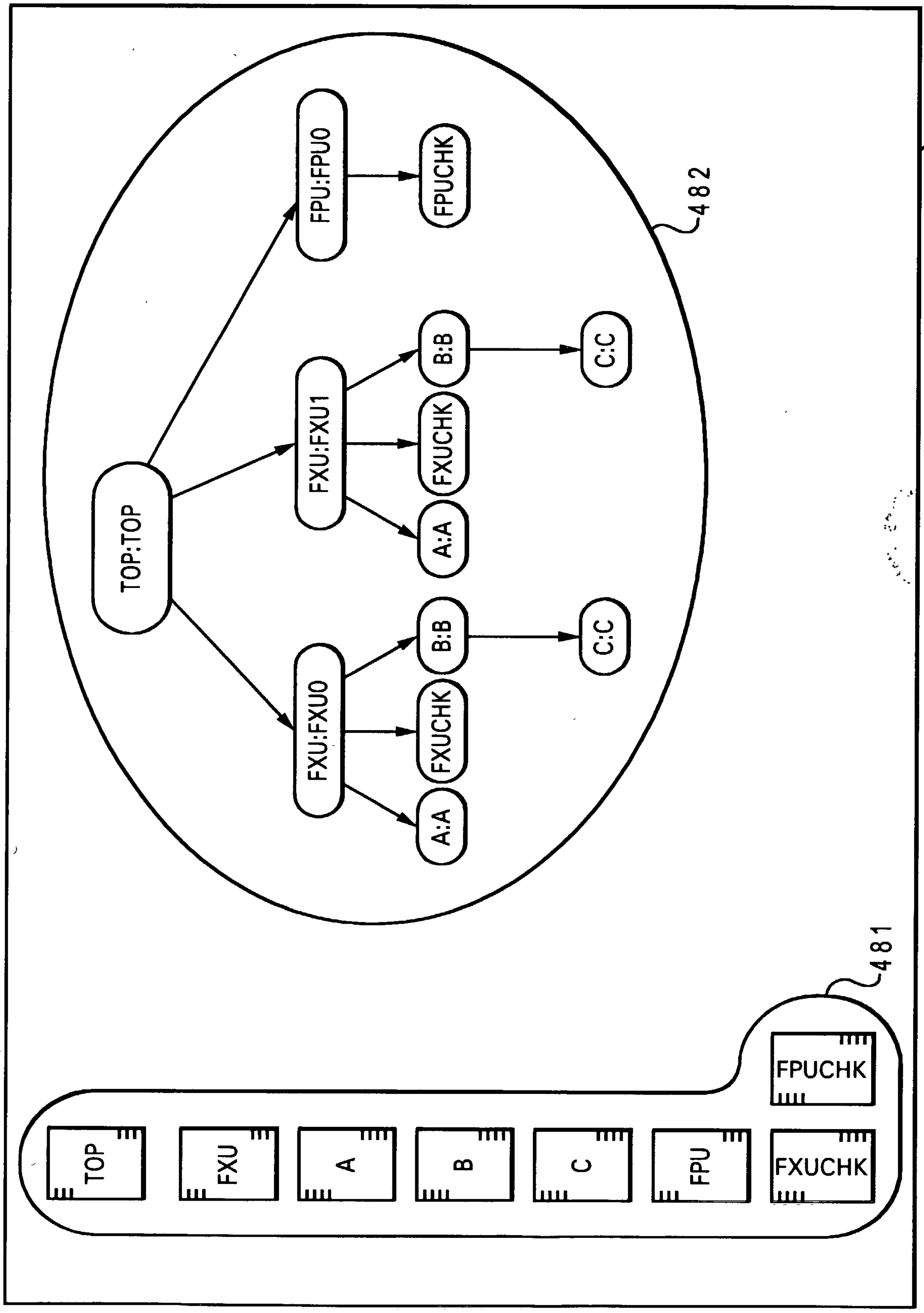

TOP:TOP
FXU:FXU0
FXU:FXU1
FPU:FPU0
A:A
FXUCHK
B:B
C:C
A:A
FXUCHK
B:B
C:C
FPUCHK
482
44
481
TOP
FXU
A
B
C
FPU
FXUCHK
FPUCHK

*Fig. 4E*

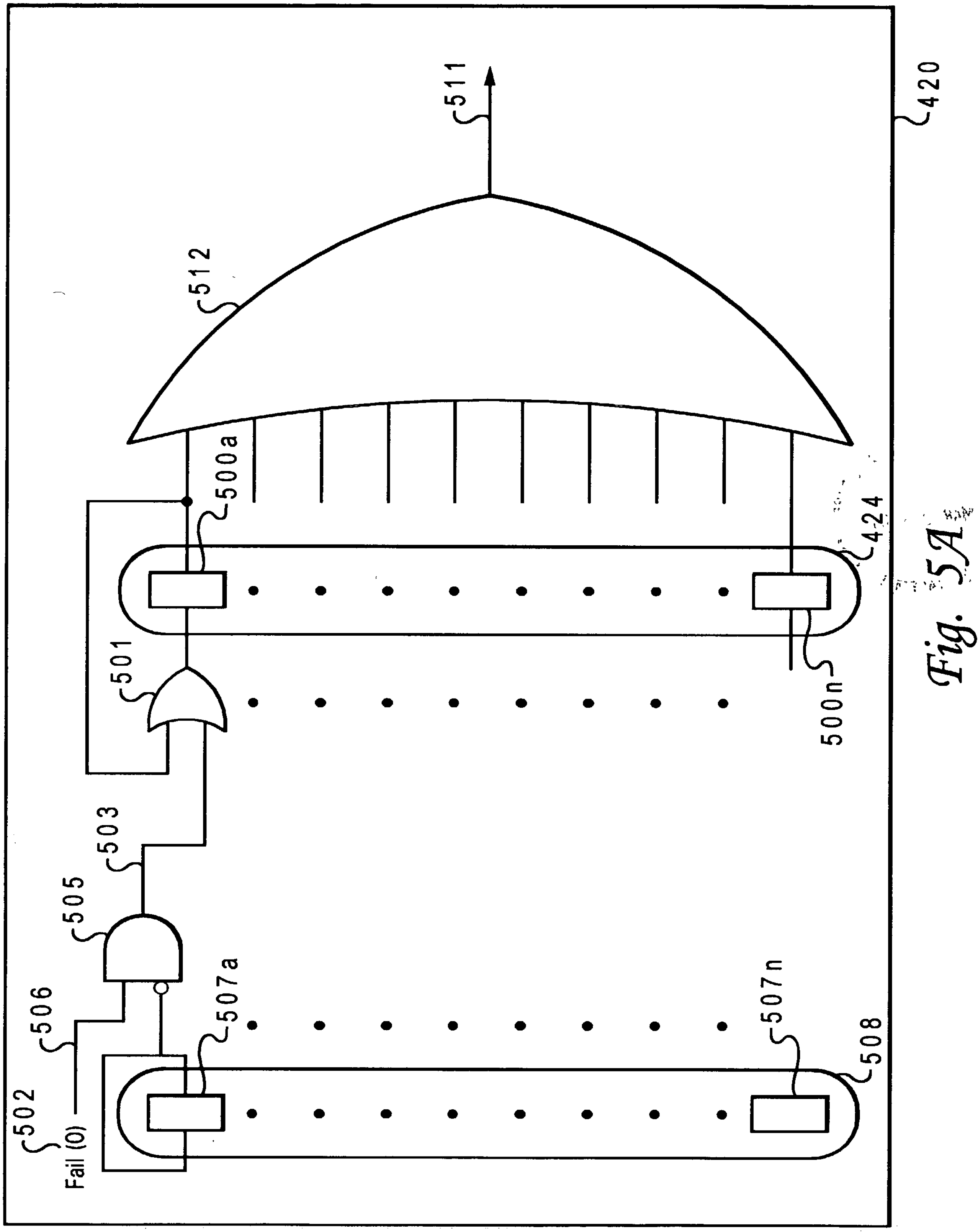

511
420
512
500a
424
501
500n
503
505
507a
507n
506
502
Fail (0)
508

Fig. 5A

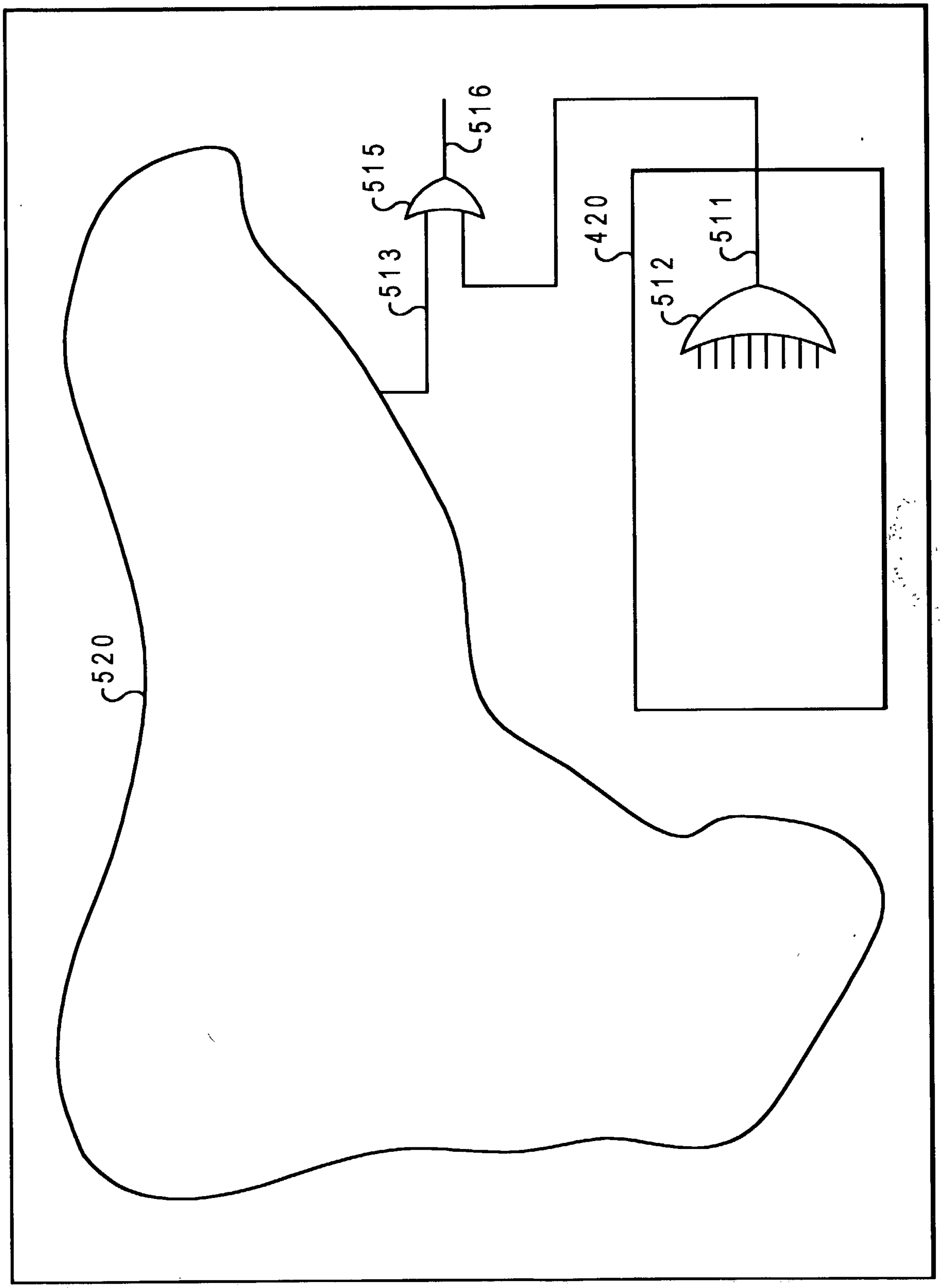


*Fig. 5B*

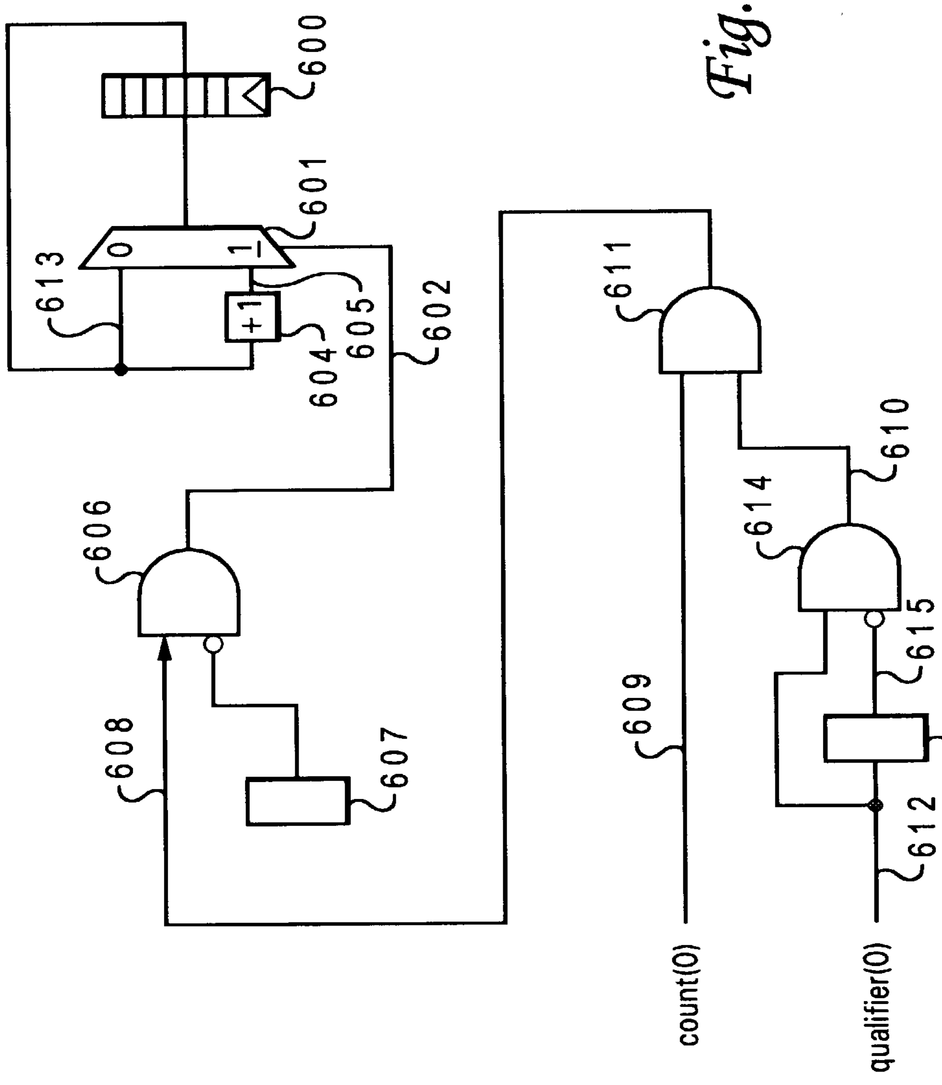

600
601
0
1
613
605
604
+1
602
606
608
607
611
610
614
615
613
612
609
count(0)
qualifier(0)

Fig. 6A

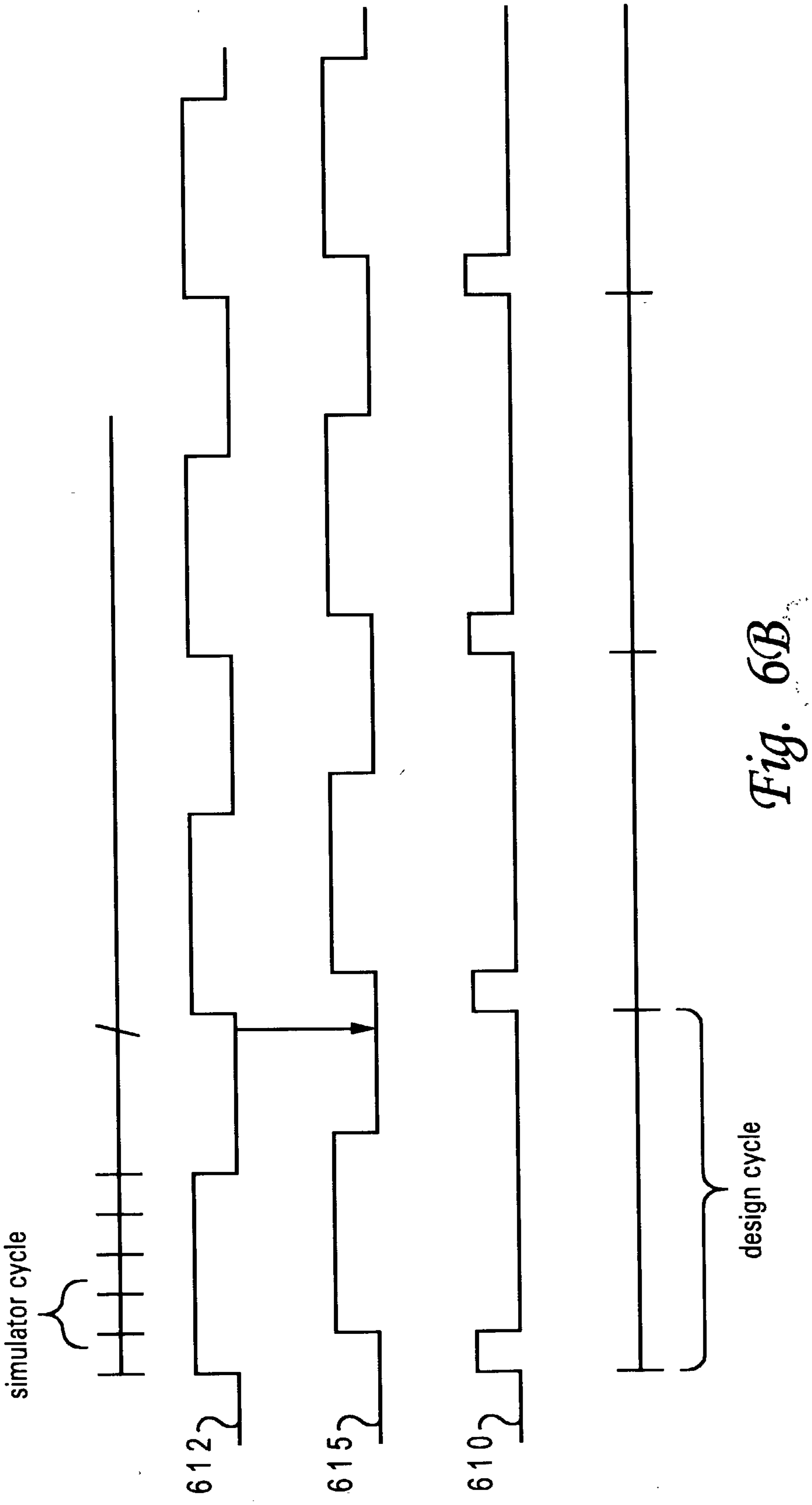


Fig. 6B

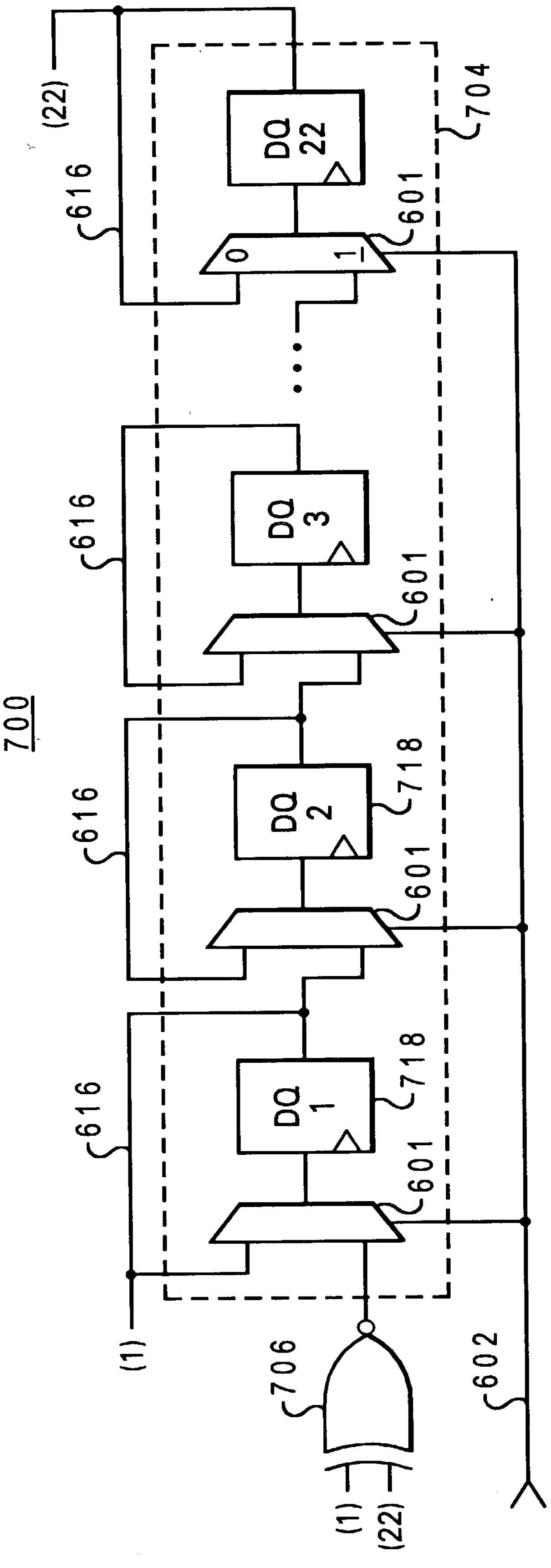

(22)
616
704
DQ
22
601
0
1
616
DQ
3
601
700
616
DQ
2
718
601
616
DQ
1
718
601
(1)
706
(1)
(22)
602

*Fig. 7*

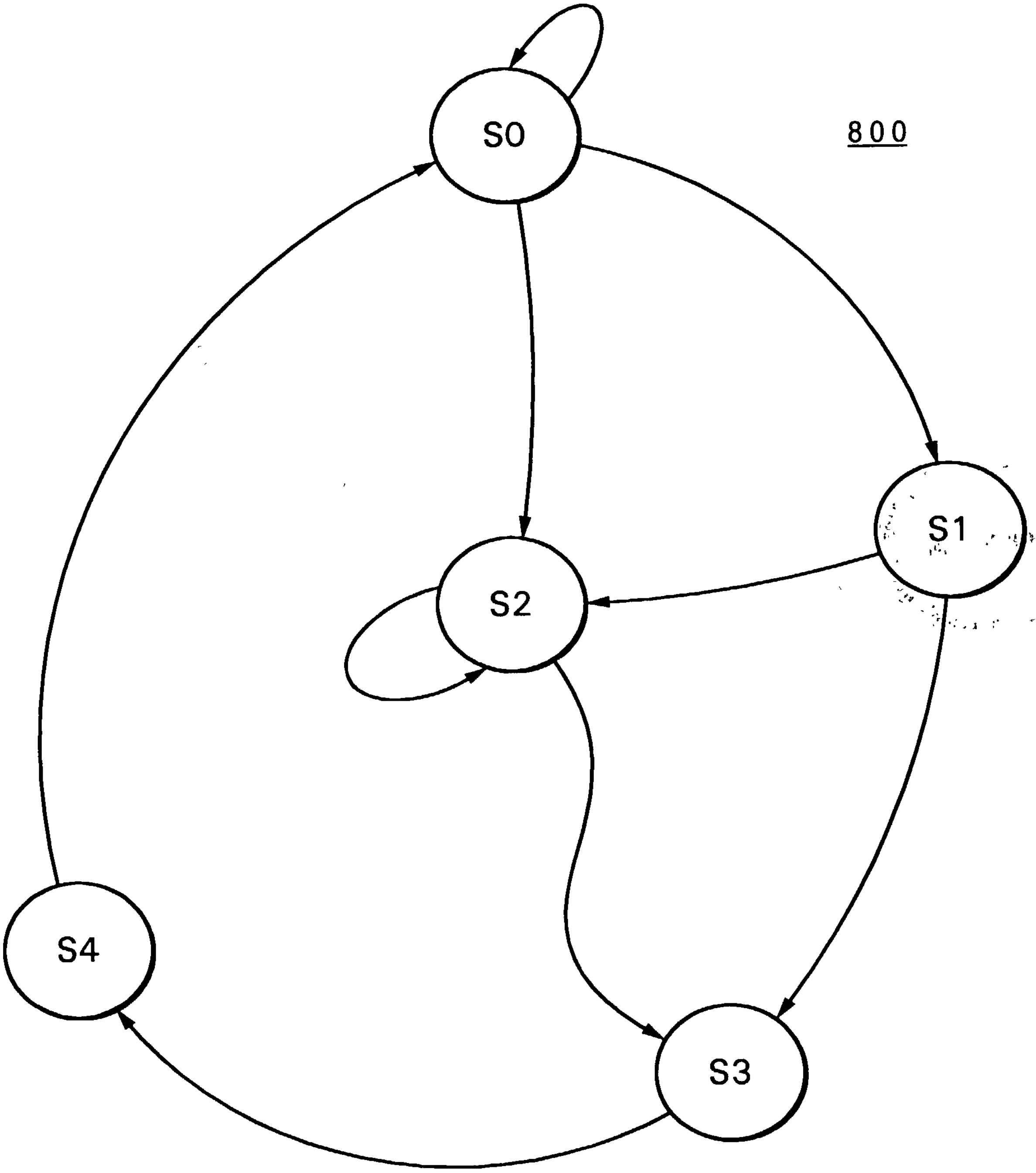


*Fig. 8A*

*Prior Art*

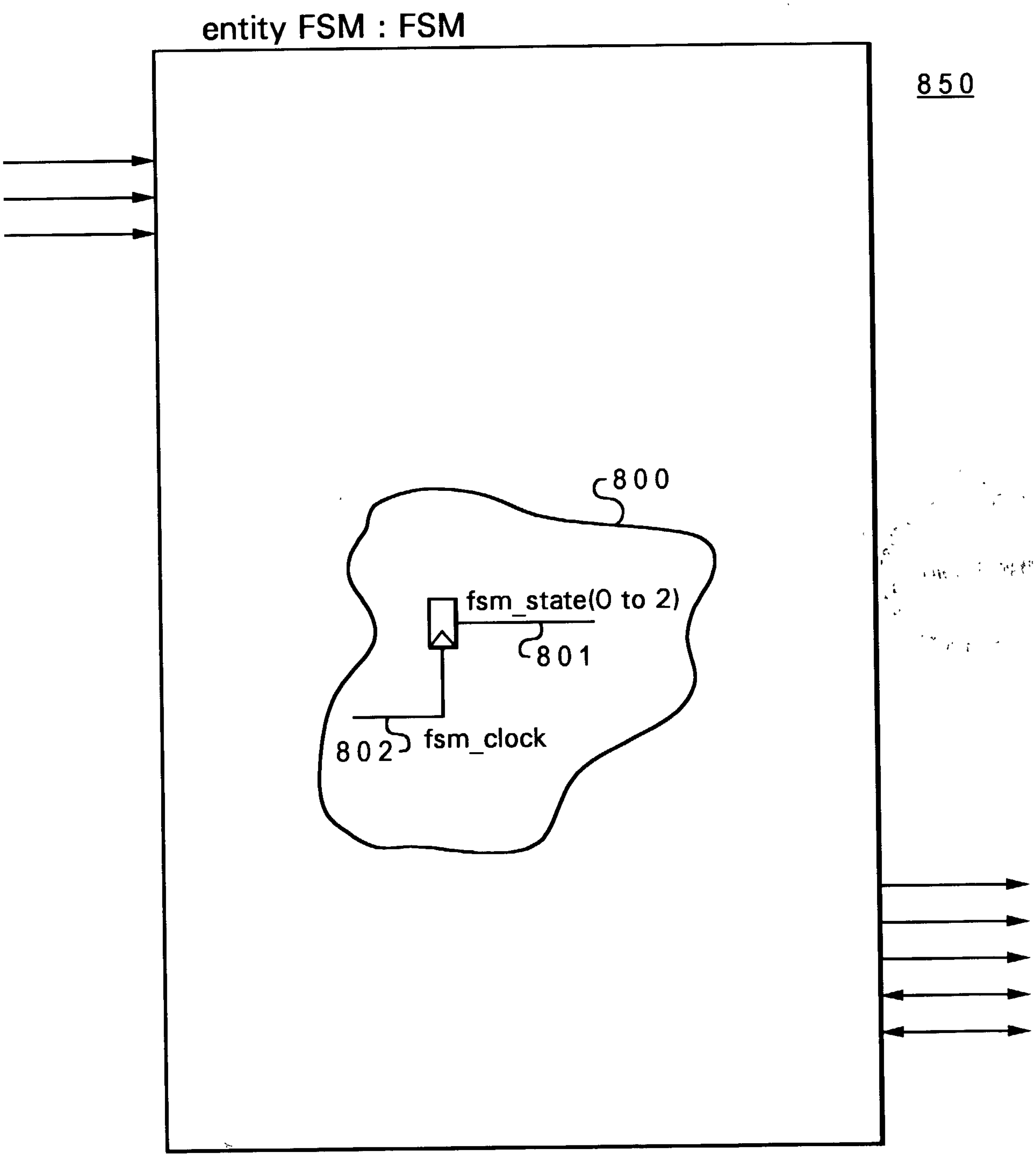


*Fig. 8B*

*Prior Art*

```
ENTITY FSM IS

     PORT(
               ....ports for entity fsm....
          );

     ARCHITECTURE FSM OF FSM IS

     BEGIN

               ... HDL code for FSM and rest of the entity ...

               fsm_state(0 to 2) <= ... Signal 801 ...

853 --!! Embedded FSM  : examplefsm;
859 --!! clock         : (fsm_clock);
854 --!! state_vector  : (fsm_state(0 to 2));
855 --!! states        : (S0, S1, S2, S3, S4);
856 --!! state_encoding : ('000', '001', '010', '011', '100');      852   860
    --!! arcs          : (S0 => S0, S0 => S1, S0 => S2,
857 --!!                  (S1 => S2, S1 => S3, S2 => S2,
    --!!                  (S2 => S3, S3 => S4, S4 => S0);
858 --!! End FSM;
                    ,
                    ,
                    ,
                    ,
                    ,
                    ,
                    ,

     END;
```

*Fig. 8C*

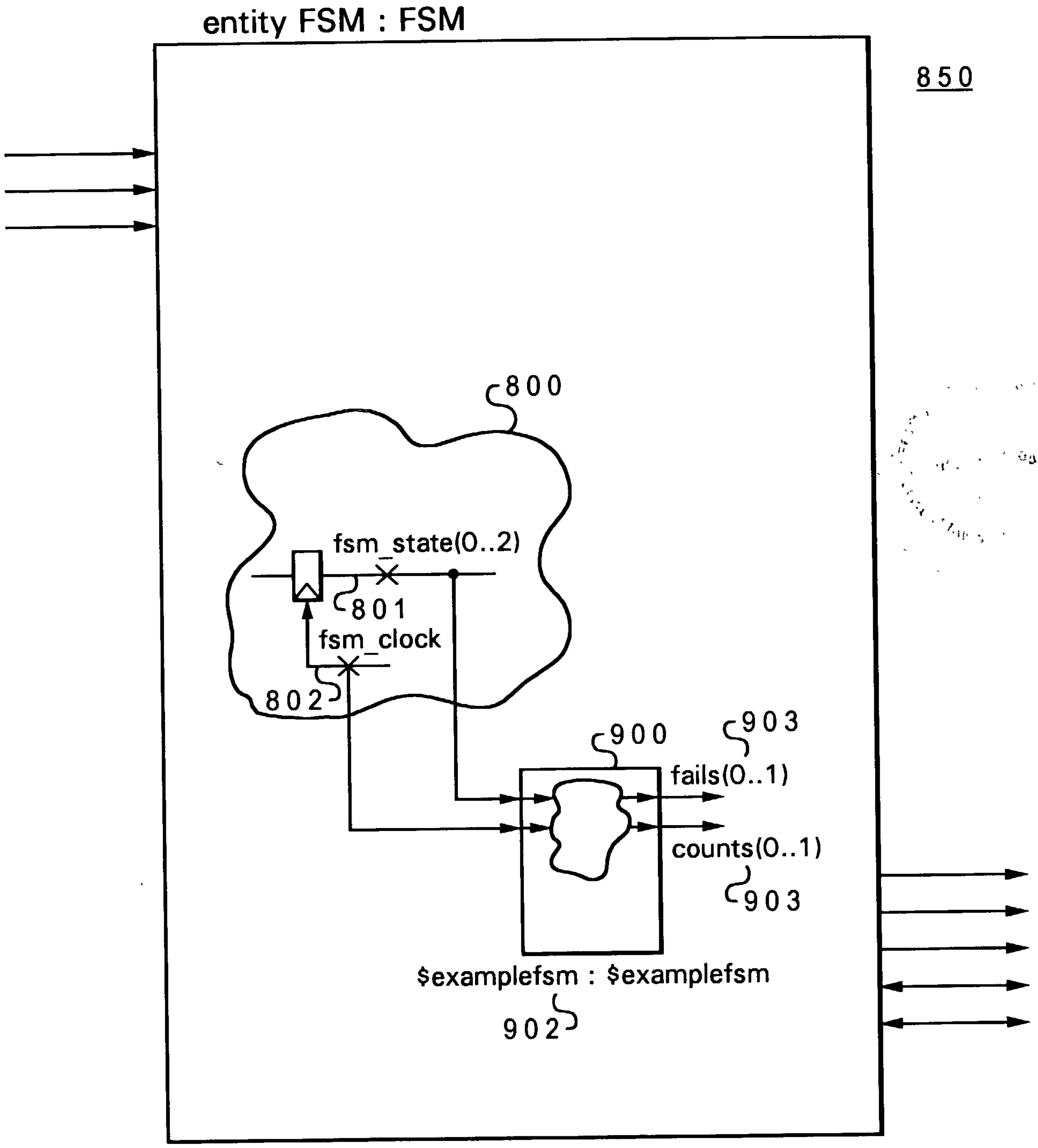
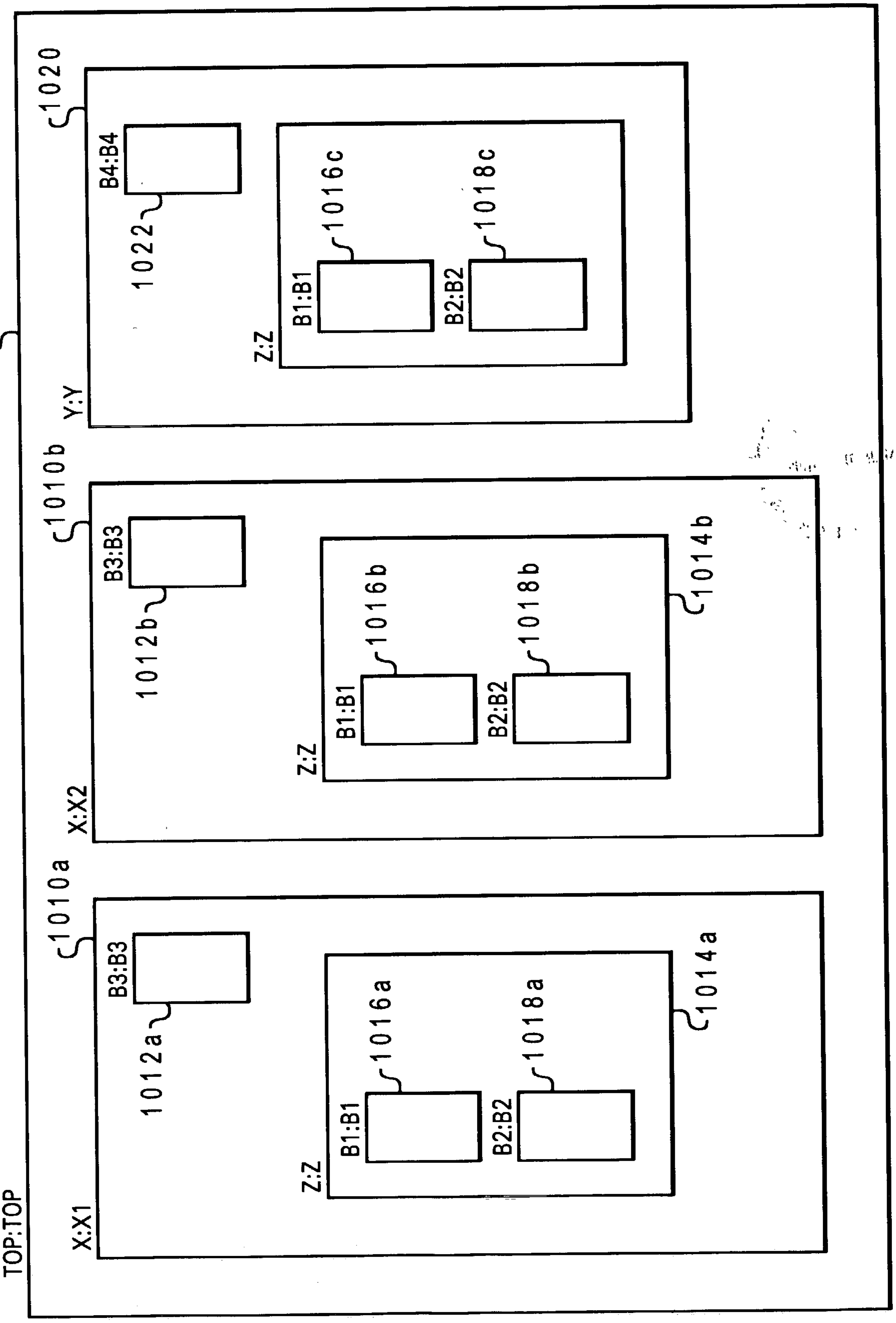

entity FSM : FSM
850
800
fsm_state(0..2)
801
fsm_clock
802
900
903
fails(0..1)
counts(0..1)
903
$examplefsm : $examplefsm
902

*Fig. 9*

Fig. 10A


1000
TOP:TOP
1020
Y:Y
B4:B4
1022
Z:Z
B1:B1
1016c
B2:B2
1018c
1010b
X:X2
B3:B3
1012b
1014b
Z:Z
B1:B1
1016b
B2:B2
1018b
1010a
X:X1
B3:B3
1012a
1014a
Z:Z
B1:B1
1016a
B2:B2
1018a

1030 1032 1034 1036

<instantiation identifier>.<instrumentation entity name>.<design entity name>.<eventname>

*Fig. 10B*

| 1030 | 1032 | 1034 | 1036 | |
|---|---|---|---|---|
| X1 | B3 | X | COUNT1 | 1040 |
| X1.Z | B1 | Z | COUNT1 | 1041 |
| X1.Z | B2 | Z | COUNT1 | 1042 |
| X2 | B3 | X | COUNT1 | 1043 |
| X2.Z | B1 | Z | COUNT1 | 1044 |
| X2.Z | B2 | Z | COUNT1 | 1045 |
| Y | B4 | Y | COUNT1 | 1046 |
| Y.Z | B1 | Z | COUNT1 | 1047 |
| Y.Z | B2 | Z | COUNT1 | 1048 |

*Fig. 10C*

1030 1034 1036

<instantiation identifier>.<design entity name>.<eventname>

*Fig. 10D*

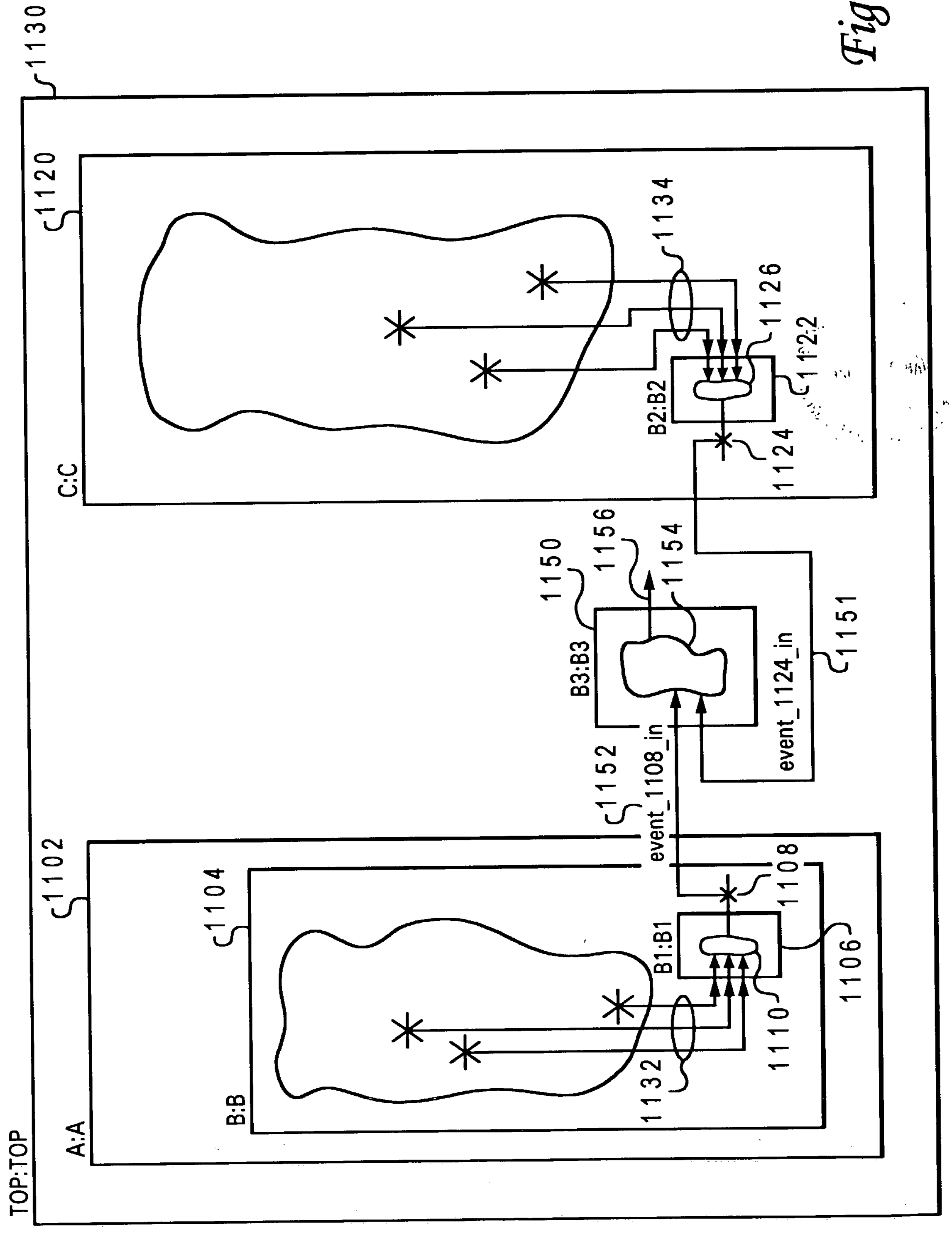
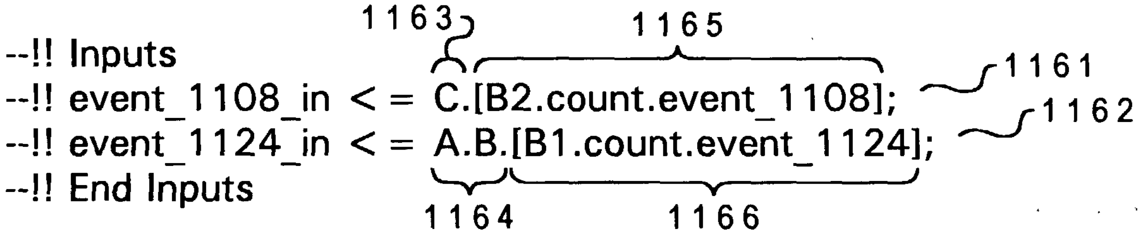
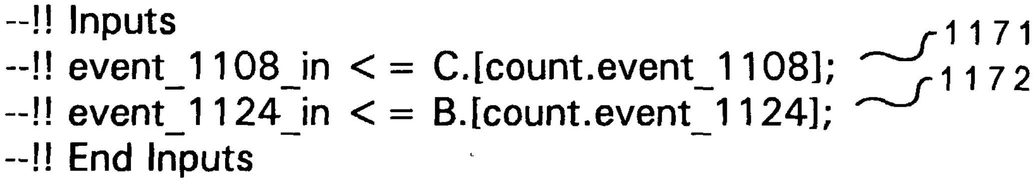

1130
1120
1134
1126
1122
B2:B2
1124
C:C
1150
1156
1154
B3:B3
1151
event_1124_in
1152
event_1108_in
1102
1104
1108
B1:B1
1106
1110
1132
B:B
A:A
TOP:TOP

Fig. 11A



```
--!! Inputs
--!! event_1108_in < = C.[B2.count.event_1108];
--!! event_1124_in < = A.B.[B1.count.event_1124];
--!! End Inputs
```

*Fig. 11B*



```
--!! Inputs
--!! event_1108_in < = C.[count.event_1108];
--!! event_1124_in < = B.[count.event_1124];
--!! End Inputs
```

*Fig. 11C*

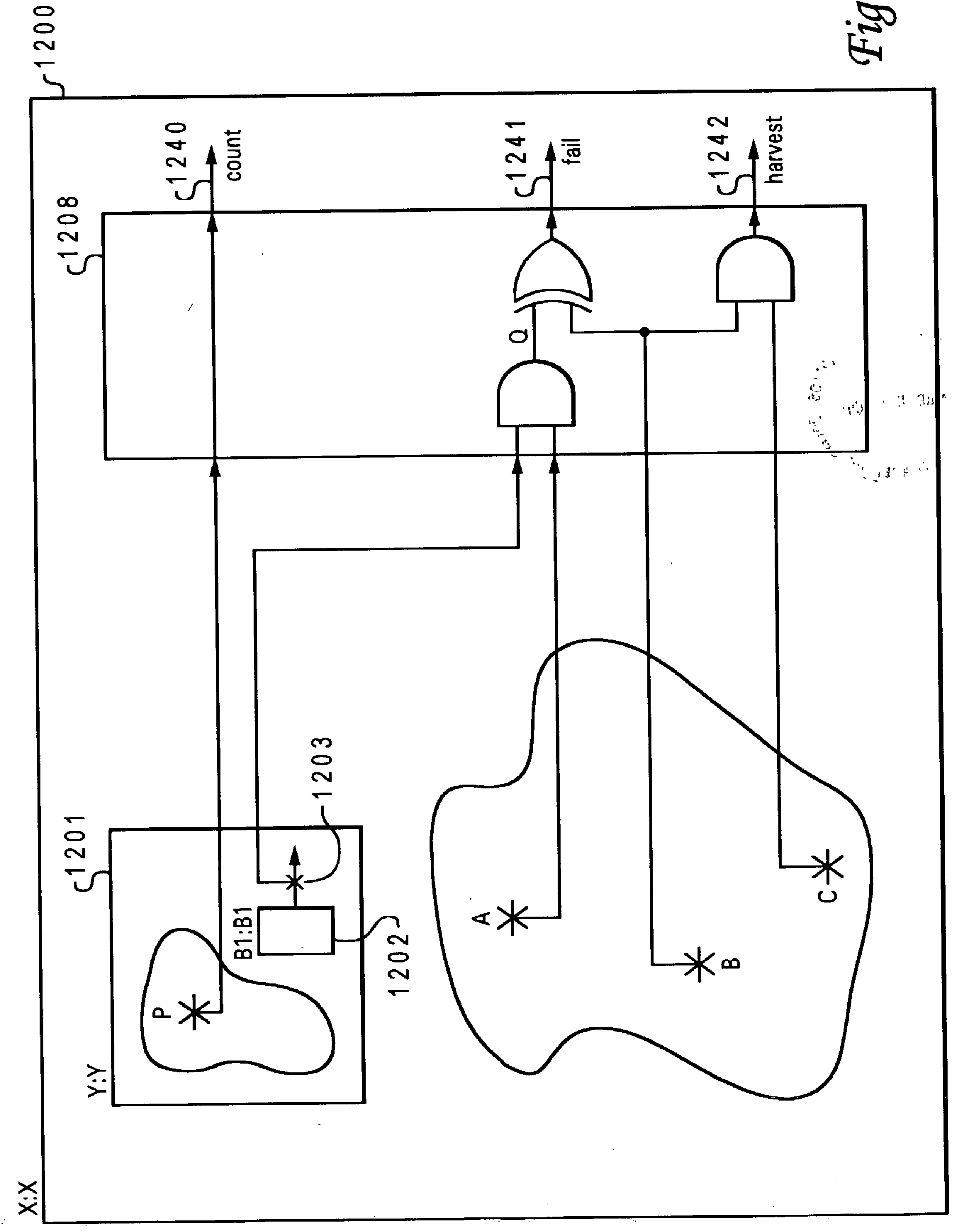

1200
X:X
1201
Y:Y
P
B1:B1
1202
1203
1208
1240
count
1241
fail
1242
harvest
Q
A
B
C

*Fig. 12A*

```
                ENTITY X IS
                       PORT(      :
                                  :
                                  :
                              );
                ARCHITECTURE example of X IS
                BEGIN
                       .
                       .
                       .
                       .
                       .
                    ... HDL code for X ...
                       .
                       .
                       .
                       .
                       .                                          1220
         (  Y:Y
   1221 {   PORT MAP(    :
         (               :
         (          );
         (  A <= ....
   1222 {   B <= ....
         (  C <= ....                              1230
         (  --!! [count, countname0, clock] <= Y.P;      1232
   1223 {   --!! Q <= Y. [B1.count.count1] AND A;             1234
         (  --!! [fail, failname0, "fail msg"] <= Q XOR B;
         (  --!! [harvest, harvestname0, "harvest msg"] <= B AND C;
                END;                              1236
```

*Fig. 12B*

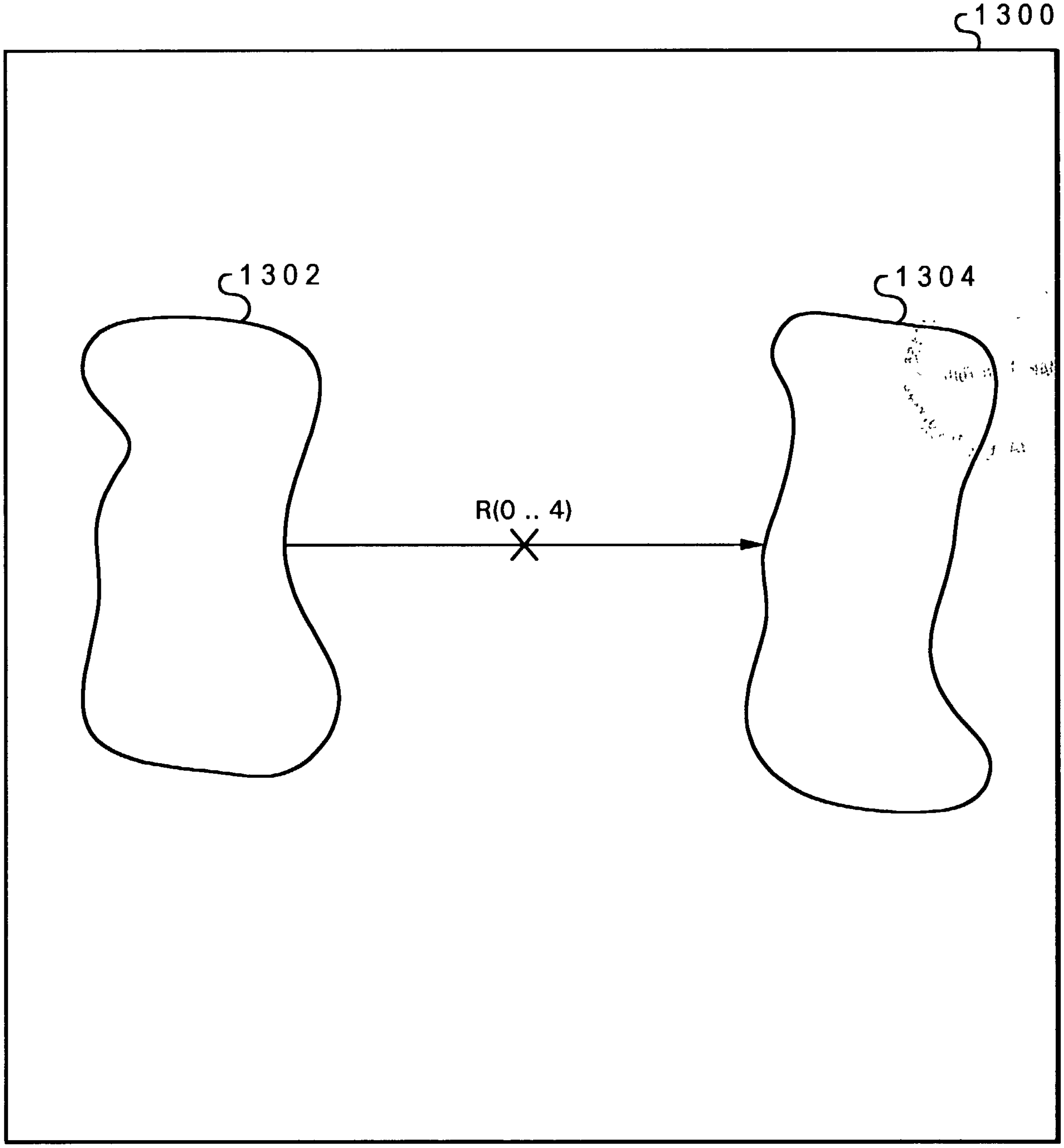


*Fig. 13A*

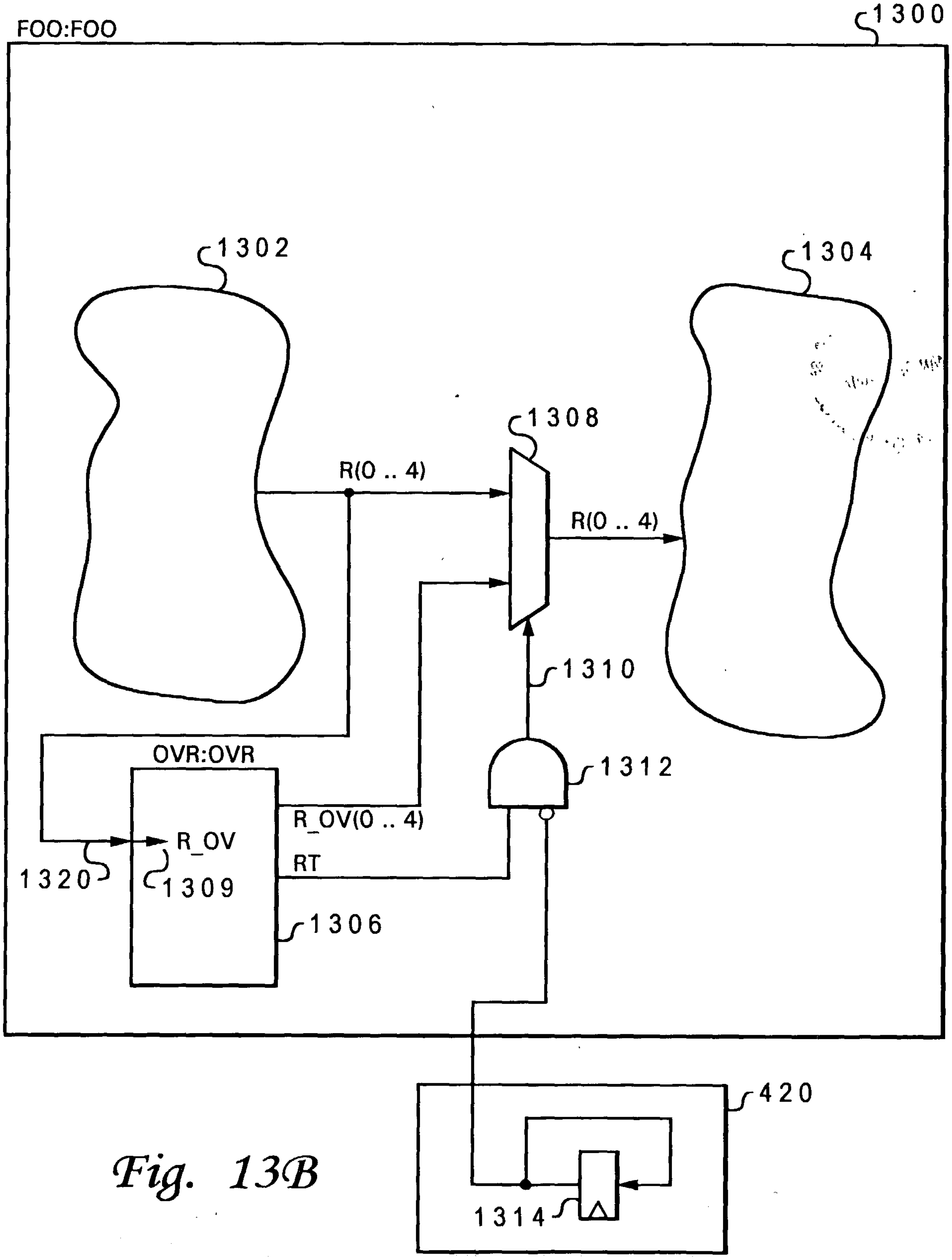

1300
FOO:FOO
1302
1304
1308
R(0 .. 4)
R(0 .. 4)
1310
1312
OVR:OVR
R_OV(0 .. 4)
R_OV
RT
1320
1309
1306
420
1314

*Fig. 13B*

```
ENTITY OVR IS                                  1364
      PORT(    R_IN          :       IN std_ulogic_vector(0 .. 4);
                 .
                 .
                 .
               ... other ports as required ...
                 .
                 .                                      1362
                 .
               R_OV          :       OUT std_ulogic_vector(0 .. 4);
               RT            :       OUT std_ulogic
          );                               1363
--!! BEGIN
--!! Design Entity: FOO;

--!! Inputs (0 to 4)                1360
--!! R_IN => {R(0 .. 4)};
--!! :
... other ports as needed ...                                          1351   1340
--!! :
--!! End Inputs
                                    1361
--!! Outputs
--!! <R_OVRRIDE> : R_OV(0 .. 4) => R(0 .. 4) [RT];                1356
--!! End Outputs

--!! End

ARCHITECTURE example of OVR IS

BEGIN

      ... HDL code for entity body section ...       1358

END;
```

*Fig. 13C*

```
ENTITY FOO IS

     PORT(      :
                :
                :
           );

ARCHITECTURE example of FOO IS

BEGIN

     .
     .
     .
     .
     .
     R <= ......
     .
     .
     .
     .
     .            1381
      --!! R_IN <= {R};
      --!!            1382
      --!!
1380  --!! R_OV(0 to 4) <= ......;    1383
      --!! RT <= ......;
      --!! [override, R_OVRRIDE, R(0 .. 4), RT] <= R_OV(0 to 4);
                          1384
```

*Fig. 13D*

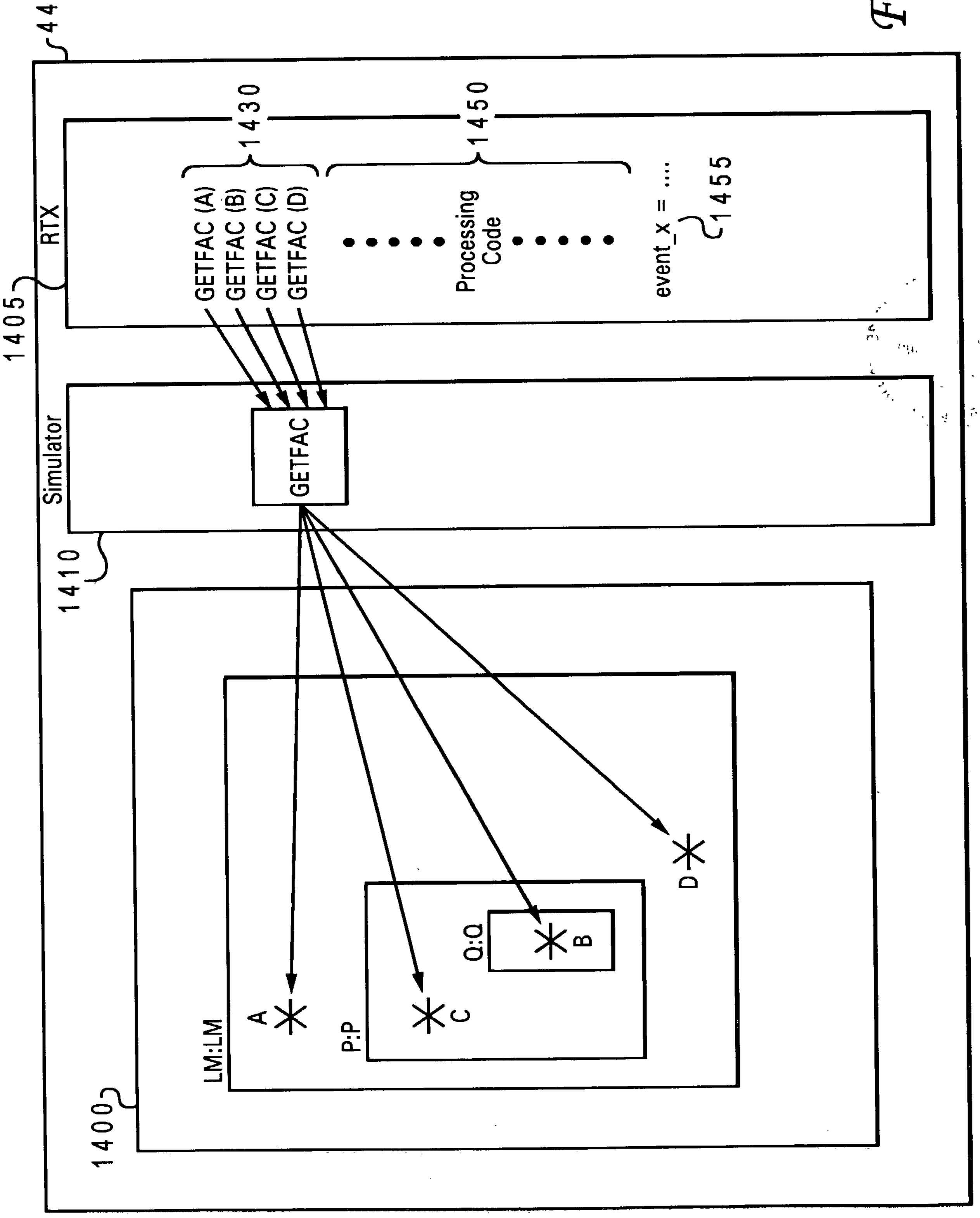

44
1405
RTX
1430
1450
GETFAC (A)
GETFAC (B)
GETFAC (C)
GETFAC (D)
Processing Code
event_x = ....
1455
Simulator
GETFAC
1410
1400
LM:LM
A
P:P
C
Q:Q
B
D

Fig. 14A

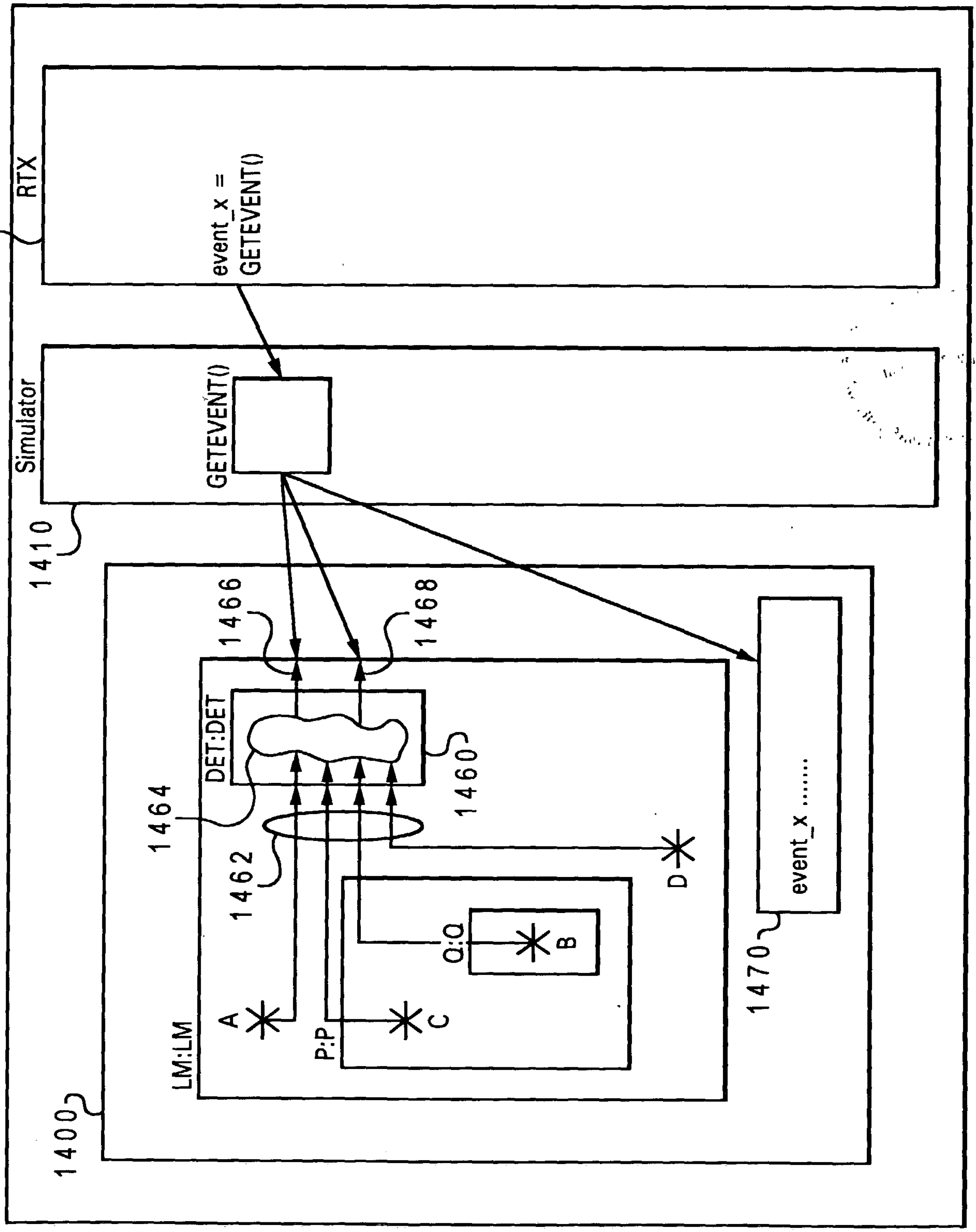
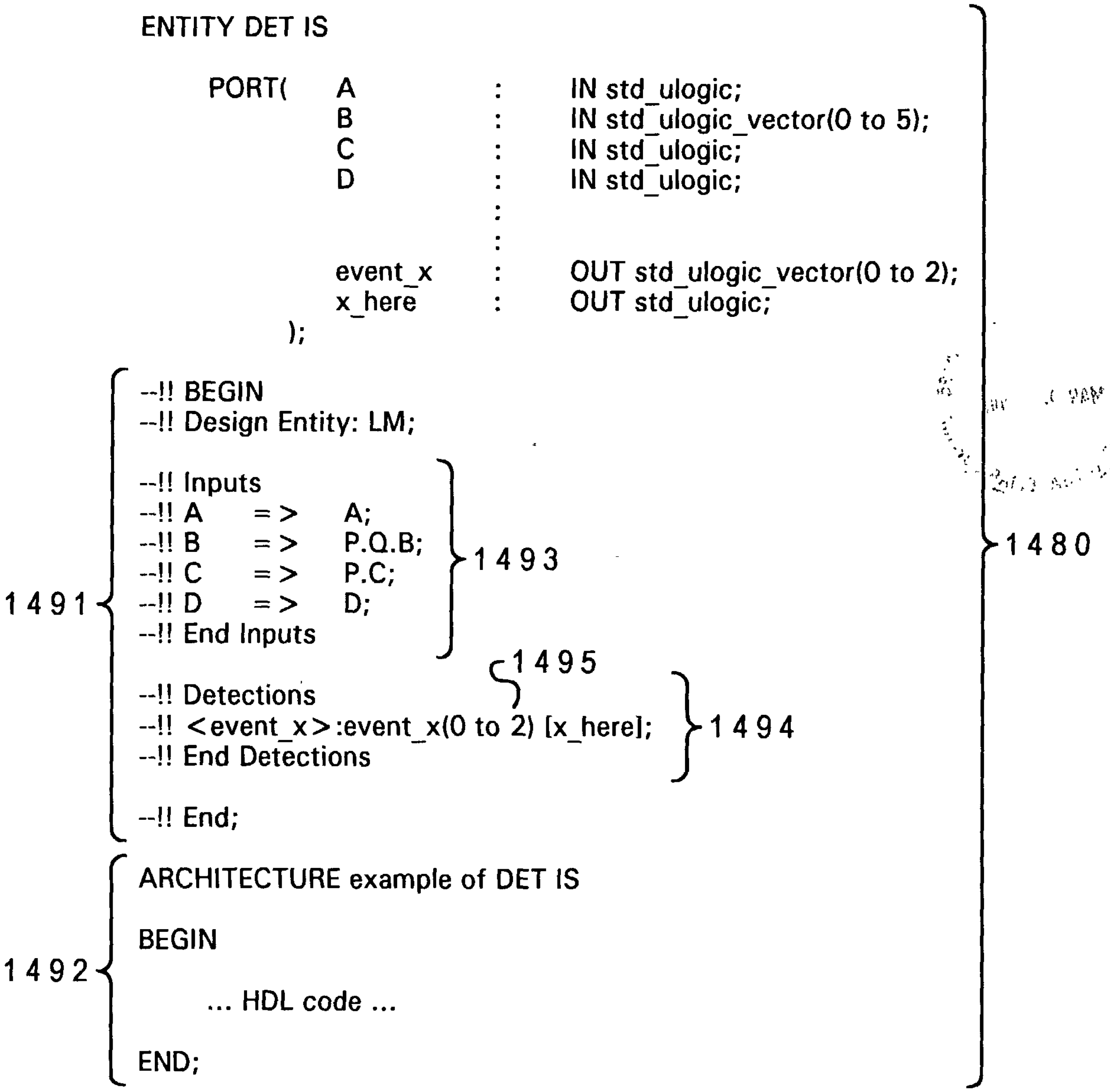
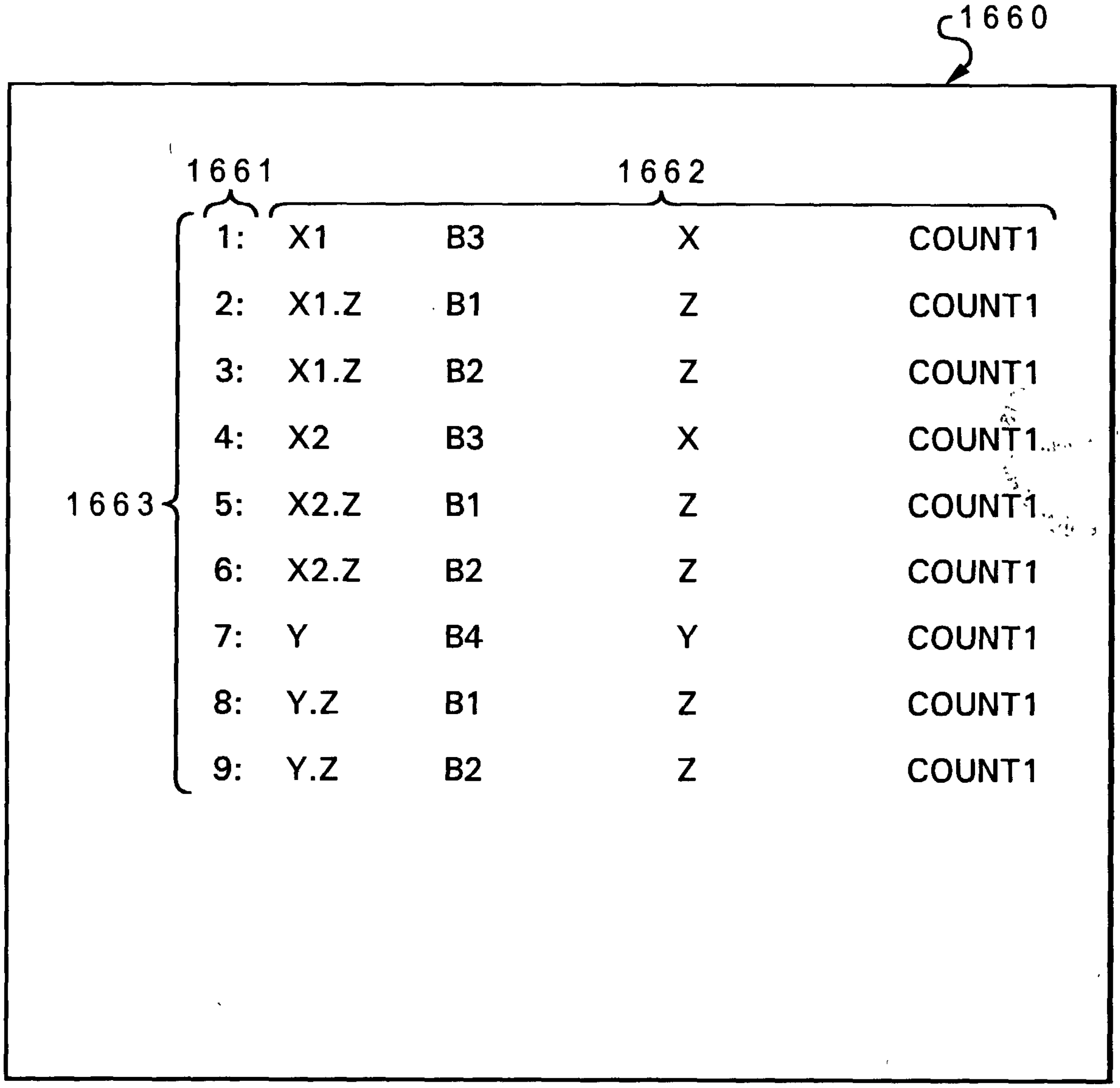

1405
RTX
event_x =
GETEVENT()
1410
Simulator
GETEVENT()
1400
1466
1468
DET:DET
1460
1464
1462
LM:LM
A
P:P
C
Q:Q
B
D
1470
event_x .......

Fig. 14B



```
ENTITY DET IS

        PORT(   A           :      IN std_ulogic;
                B           :      IN std_ulogic_vector(0 to 5);
                C           :      IN std_ulogic;
                D           :      IN std_ulogic;
                            :
                            :
                event_x     :      OUT std_ulogic_vector(0 to 2);
                x_here      :      OUT std_ulogic;
              );
--!! BEGIN
--!! Design Entity: LM;

--!! Inputs
--!! A      =>     A;
--!! B      =>     P.Q.B;
--!! C      =>     P.C;
--!! D      =>     D;
--!! End Inputs

--!! Detections
--!! <event_x>:event_x(0 to 2) [x_here];
--!! End Detections

--!! End;
ARCHITECTURE example of DET IS

BEGIN

        ... HDL code ...

END;
```

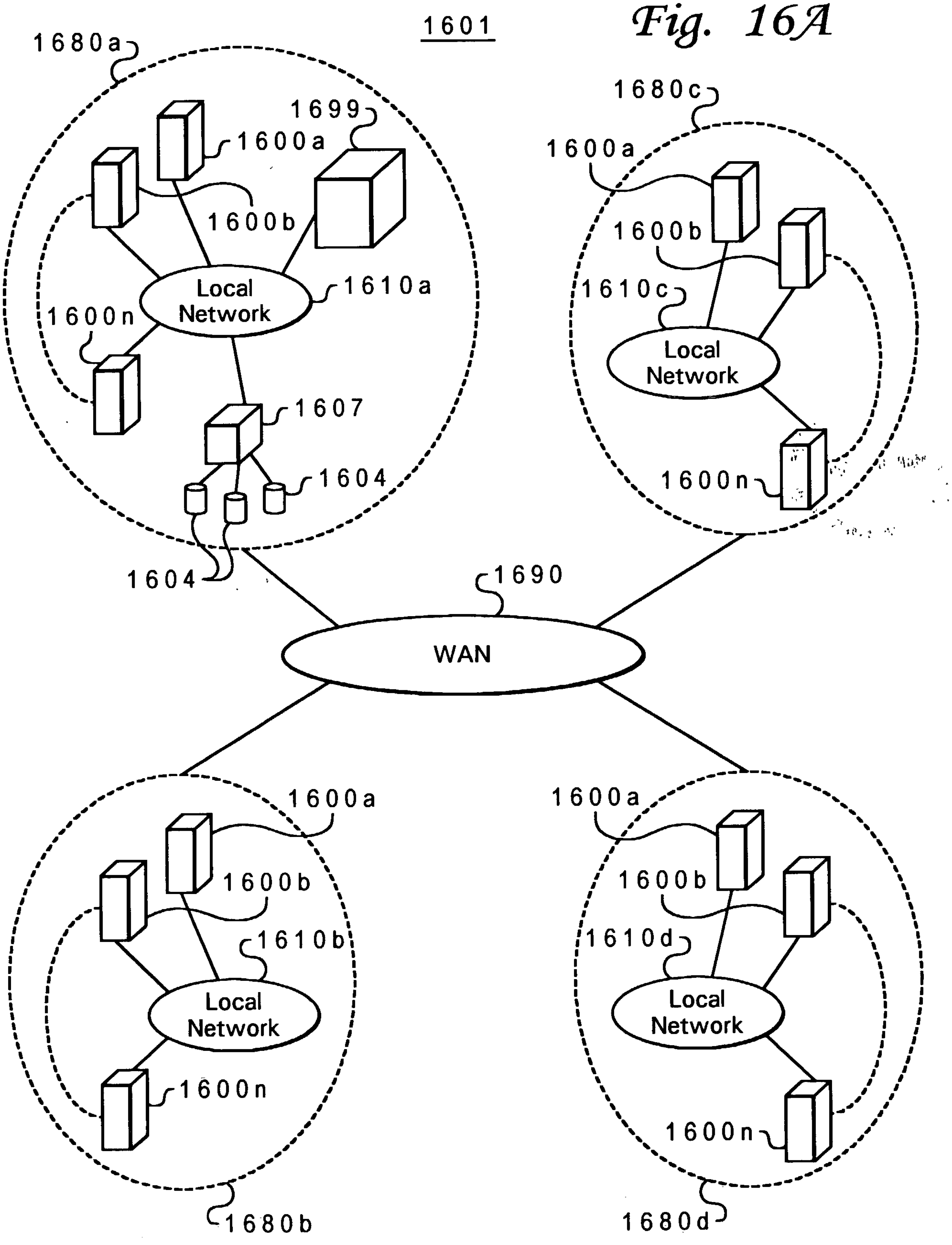
*Fig. 14C*


1660
1661
1662
1663

| | | | | |
|---|---|---|---|---|
| 1: | X1 | B3 | X | COUNT1 |
| 2: | X1.Z | B1 | Z | COUNT1 |
| 3: | X1.Z | B2 | Z | COUNT1 |
| 4: | X2 | B3 | X | COUNT1 |
| 5: | X2.Z | B1 | Z | COUNT1 |
| 6: | X2.Z | B2 | Z | COUNT1 |
| 7: | Y | B4 | Y | COUNT1 |
| 8: | Y.Z | B1 | Z | COUNT1 |
| 9: | Y.Z | B2 | Z | COUNT1 |

*Fig. 15*

*Fig. 16A*


1601
1680a
1699
1600a
1600b
Local Network
1610a
1600n
1607
1604
1604
1680c
1600a
1600b
1610c
Local Network
1600n
1690
WAN
1600a
1600b
1610b
Local Network
1600n
1680b
1600a
1600b
1610d
Local Network
1600n
1680d

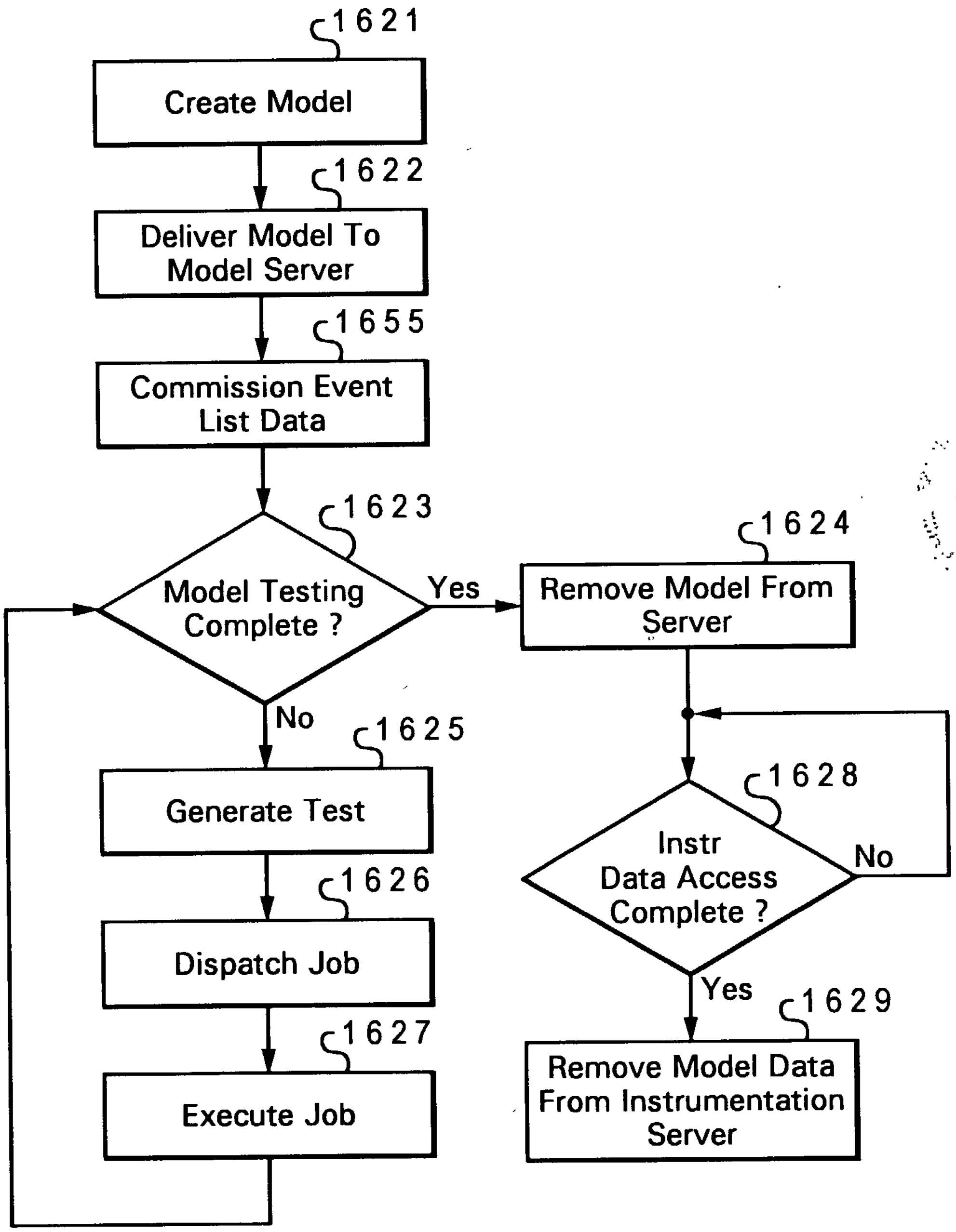

1621
Create Model
1622
Deliver Model To Model Server
1655
Commission Event List Data
1623
Model Testing Complete ?
Yes
No
1625
Generate Test
1626
Dispatch Job
1627
Execute Job
1624
Remove Model From Server
1628
Instr Data Access Complete ?
No
Yes
1629
Remove Model Data From Instrumentation Server

*Fig. 16B*

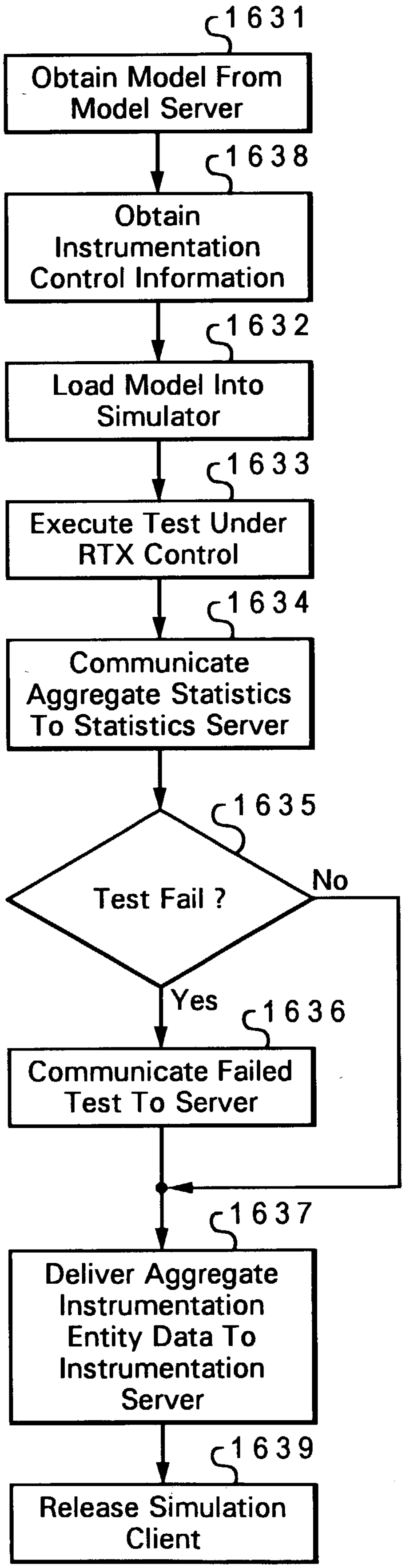

1631
Obtain Model From Model Server
1638
Obtain Instrumentation Control Information
1632
Load Model Into Simulator
1633
Execute Test Under RTX Control
1634
Communicate Aggregate Statistics To Statistics Server
1635
Test Fail ?
No
Yes
1636
Communicate Failed Test To Server
1637
Deliver Aggregate Instrumentation Entity Data To Instrumentation Server
1639
Release Simulation Client

*Fig. 16C*

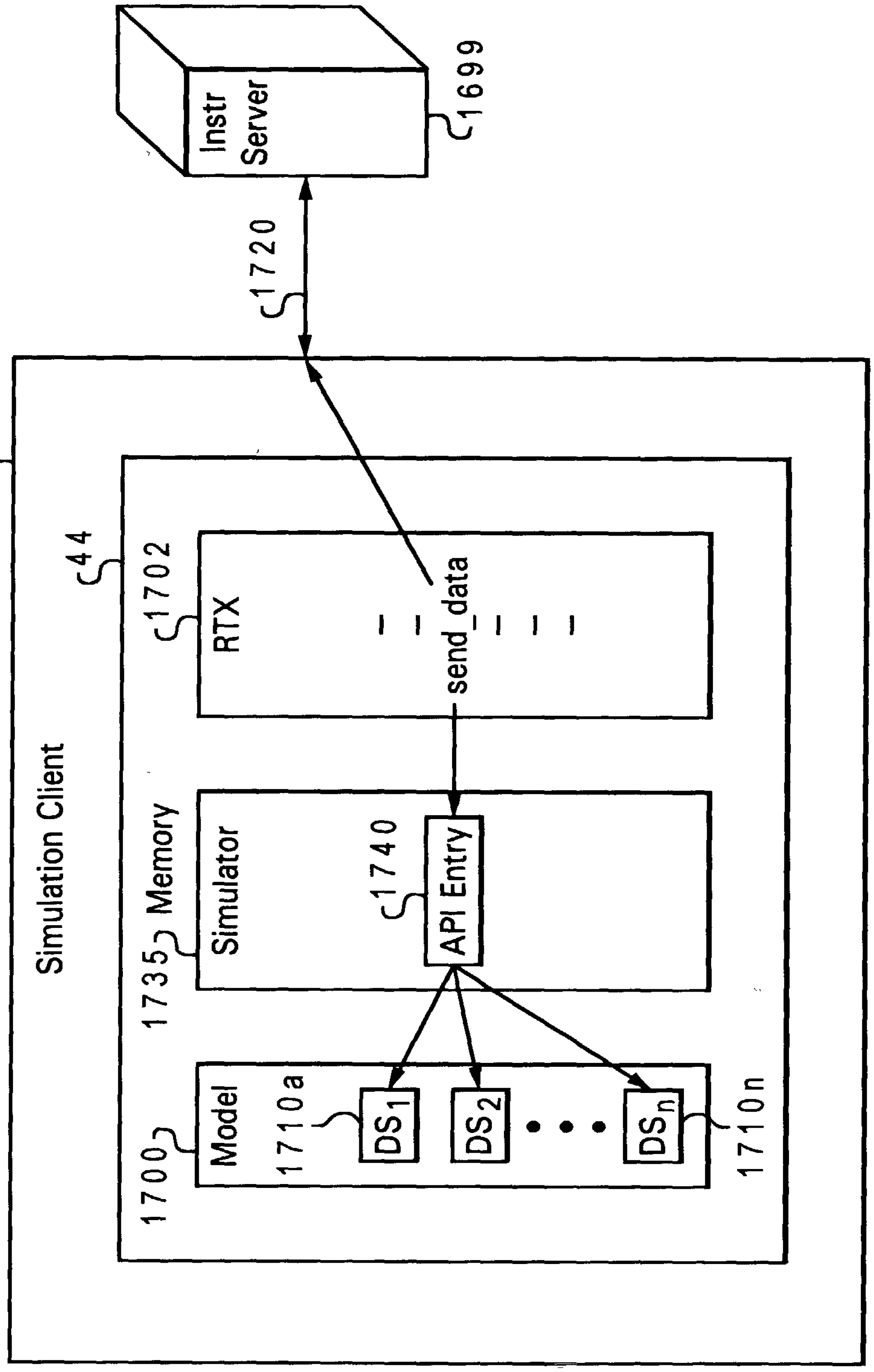

Instr Server
1699
1720
1701
Simulation Client
44
Memory
1702
RTX
send_data
1735
Simulator
1740
API Entry
1700
Model
1710a
$DS_1$
$DS_2$
• • •
$DS_n$
1710n

Fig. 17A

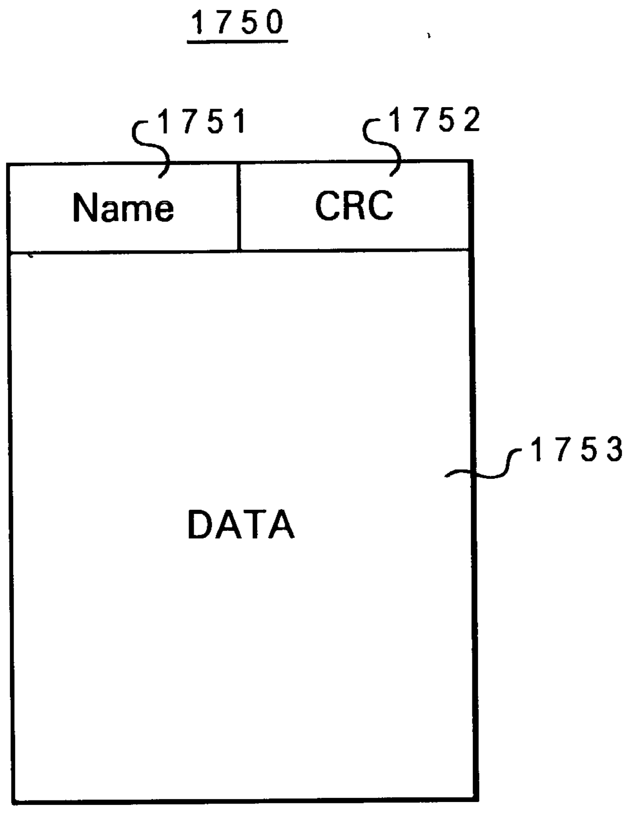


*Fig. 17B*

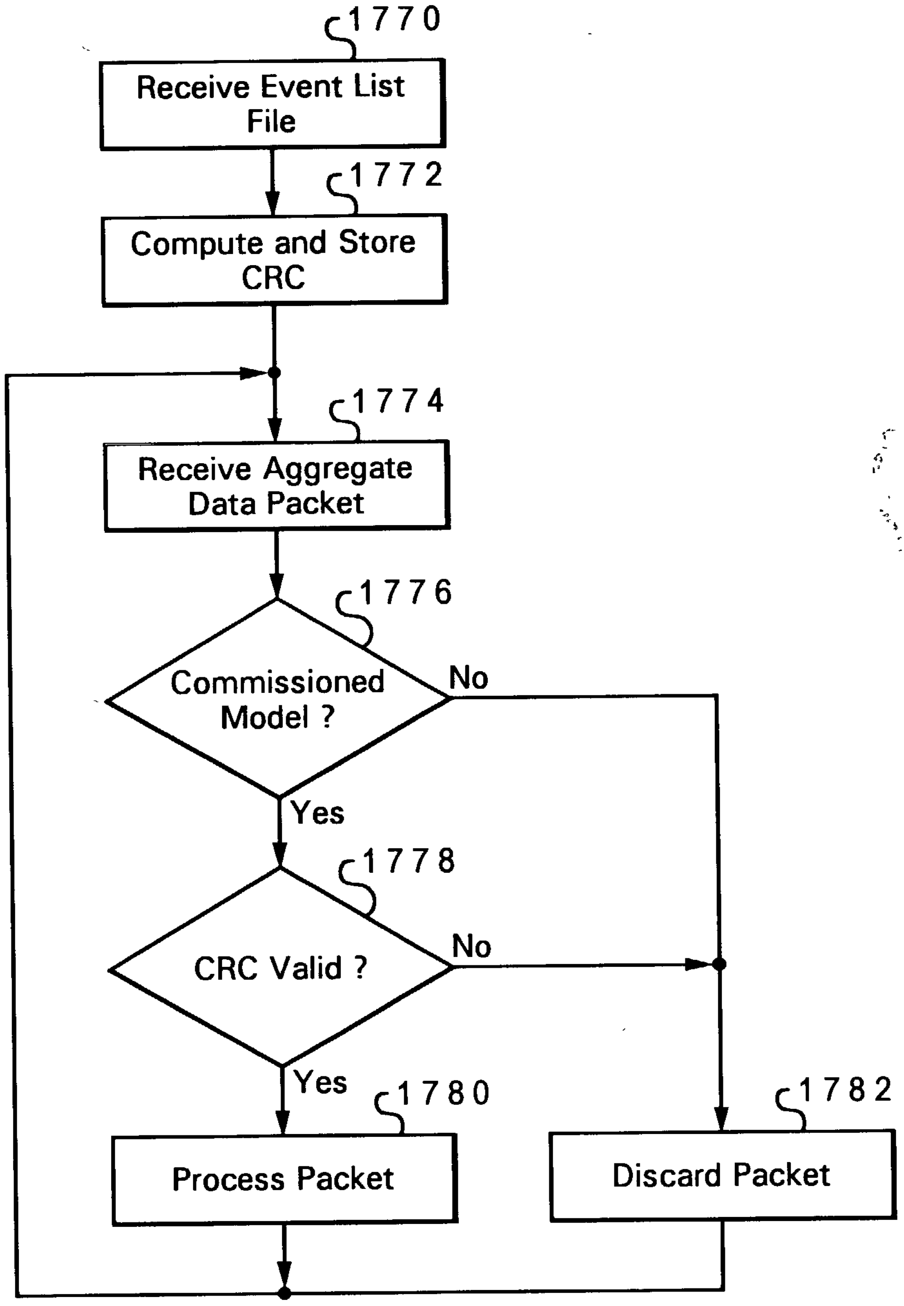

1770
Receive Event List File
1772
Compute and Store CRC
1774
Receive Aggregate Data Packet
1776
Commissioned Model ?
No
Yes
1778
CRC Valid ?
No
Yes
1780
Process Packet
1782
Discard Packet

*Fig. 17C*

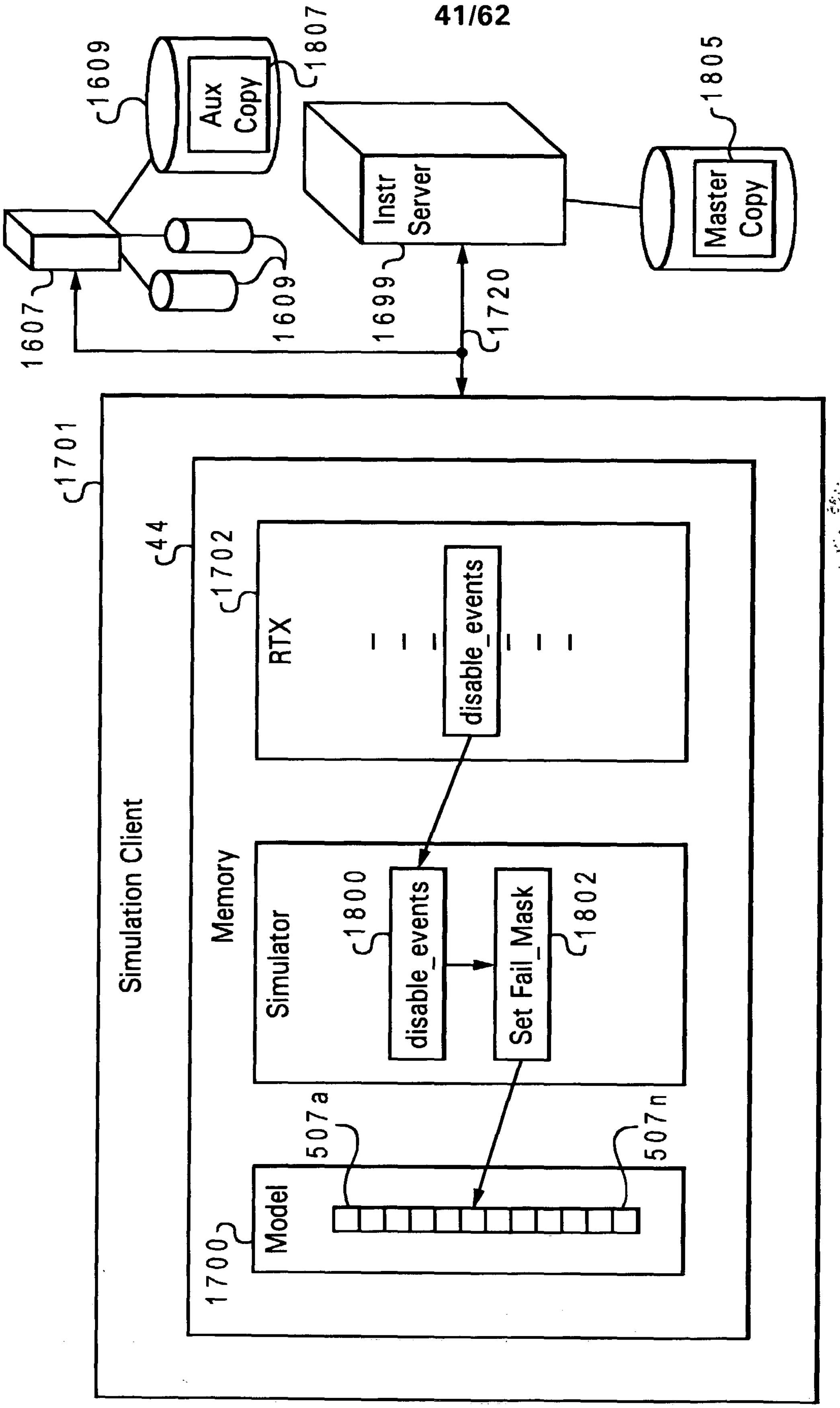

1609
Aux Copy
1807
1607
1609
1699
Instr Server
1720
1805
Master Copy
1701
Simulation Client
44
Memory
1702
RTX
disable_events
Simulator
1800
disable_events
Set Fail_Mask
1802
1700
Model
507a
507n

Fig. 18A

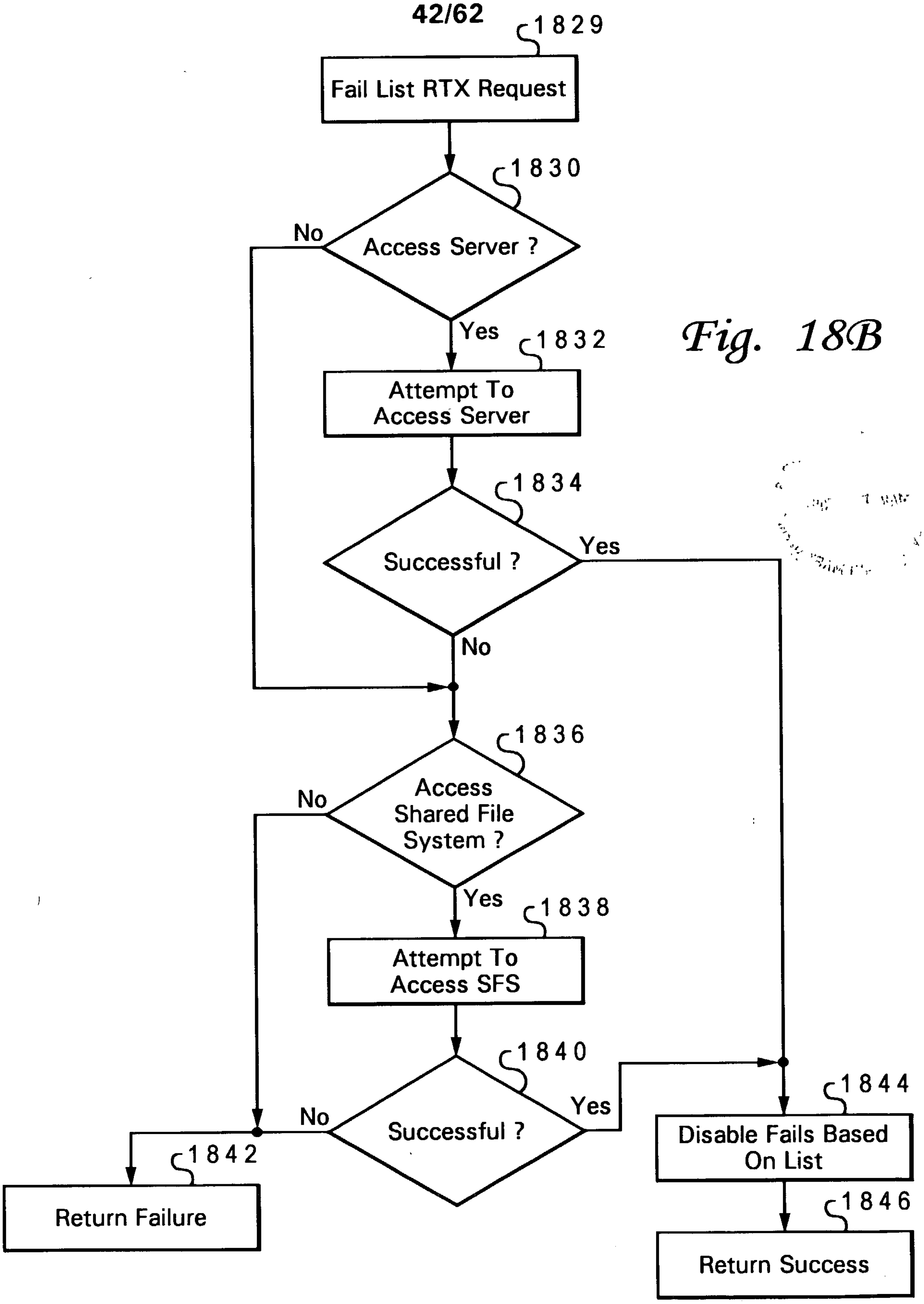

1829
Fail List RTX Request
1830
Access Server ?
No
Yes
1832
Attempt To Access Server
1834
Successful ?
Yes
No
1836
Access Shared File System ?
No
Yes
1838
Attempt To Access SFS
1840
Successful ?
No
Yes
1842
Return Failure
1844
Disable Fails Based On List
1846
Return Success

*Fig. 18B*

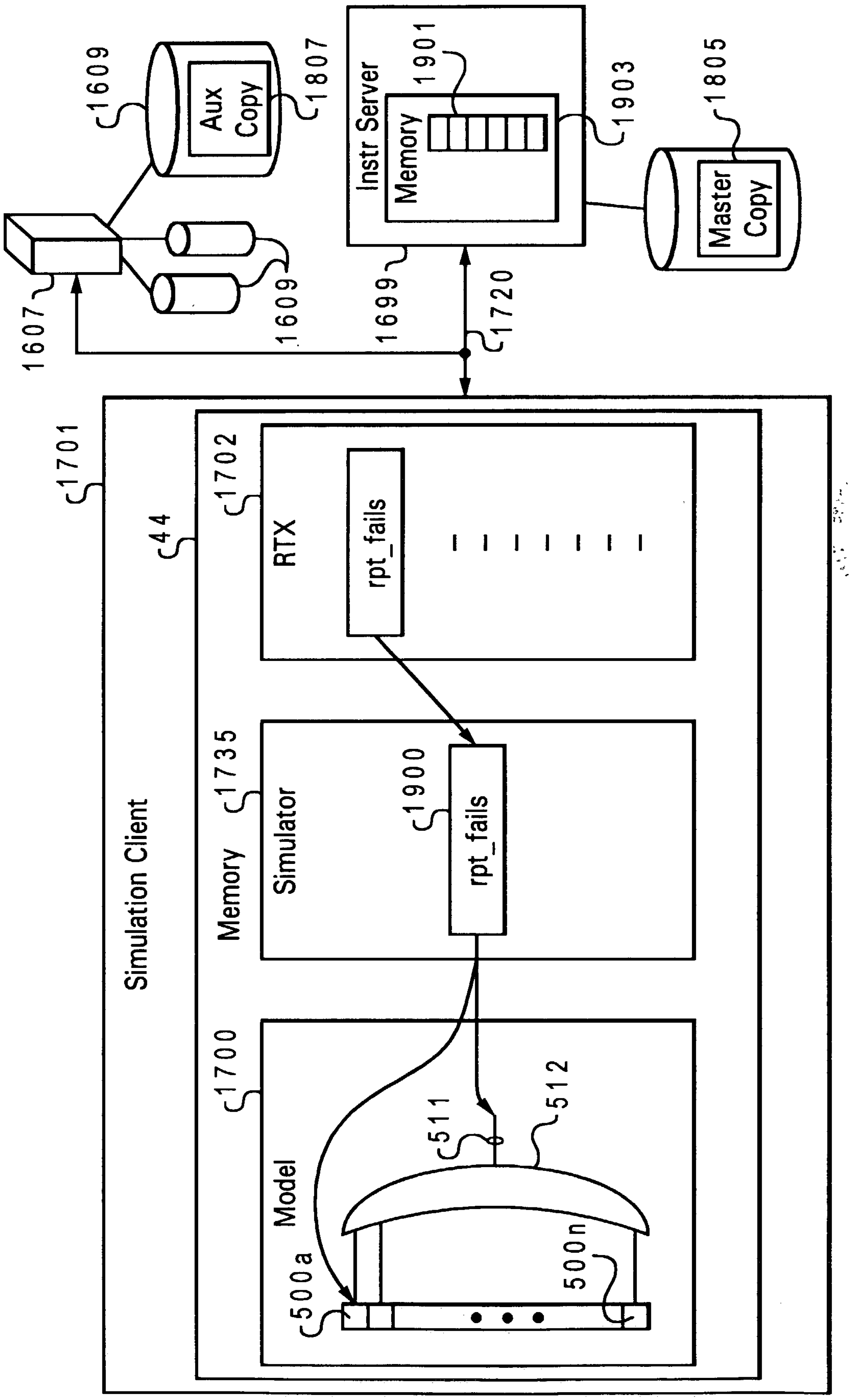

Simulation Client
1701
44
Memory
RTX
1702
rpt_fails
Simulator
1735
1900
rpt_fails
Model
1700
511
512
500a
500n
1607
1609
1609
Aux Copy
1609
1807
1699
1720
Instr Server
Memory
1901
1903
Master Copy
1805

Fig. 19A

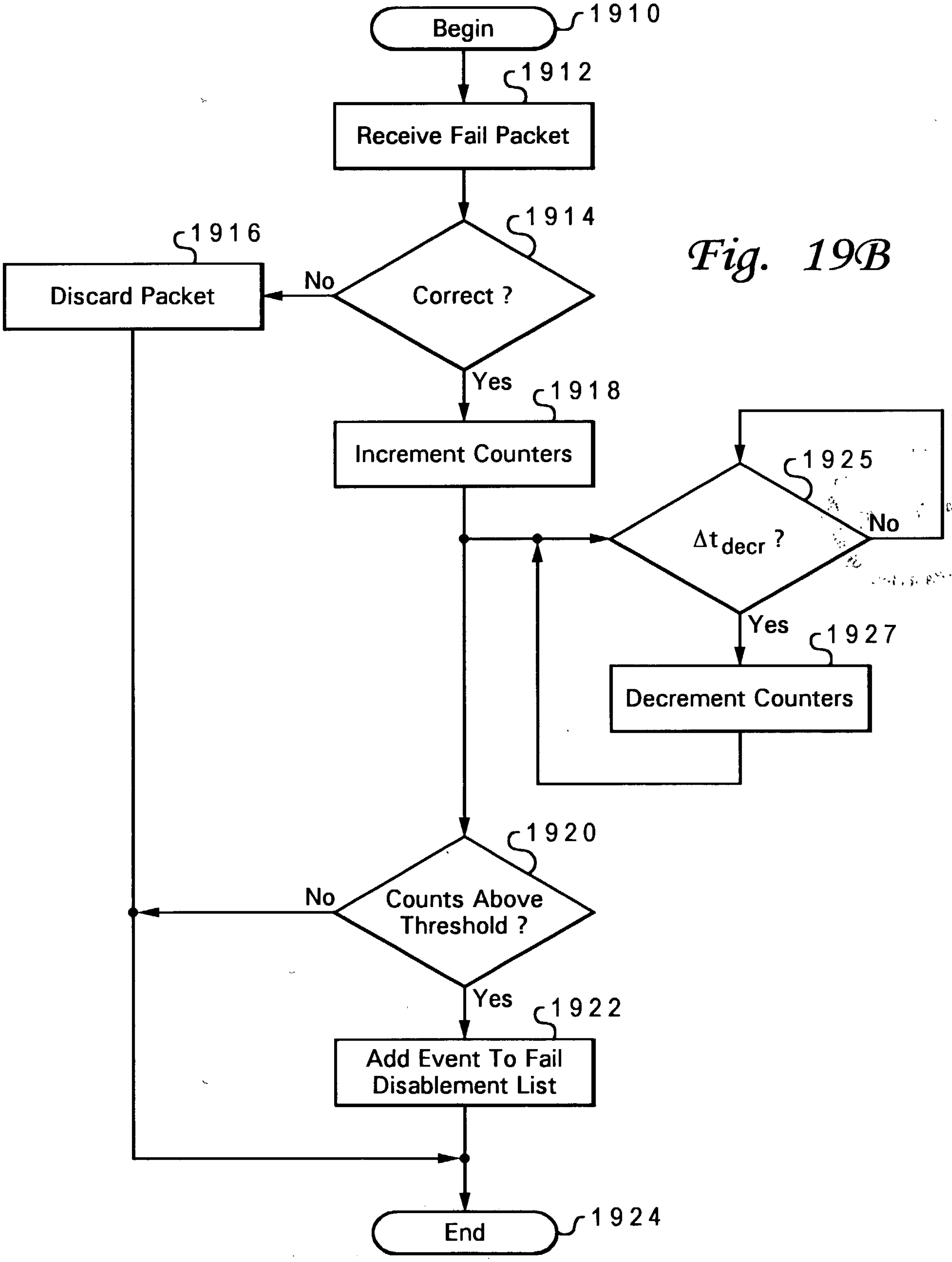

Begin
1910
Receive Fail Packet
1912
Correct ?
1914
No
Discard Packet
1916
Yes
Increment Counters
1918
$\Delta t_{decr}$ ?
1925
No
Yes
Decrement Counters
1927
Counts Above Threshold ?
1920
No
Yes
Add Event To Fail Disablement List
1922
End
1924

*Fig. 19B*

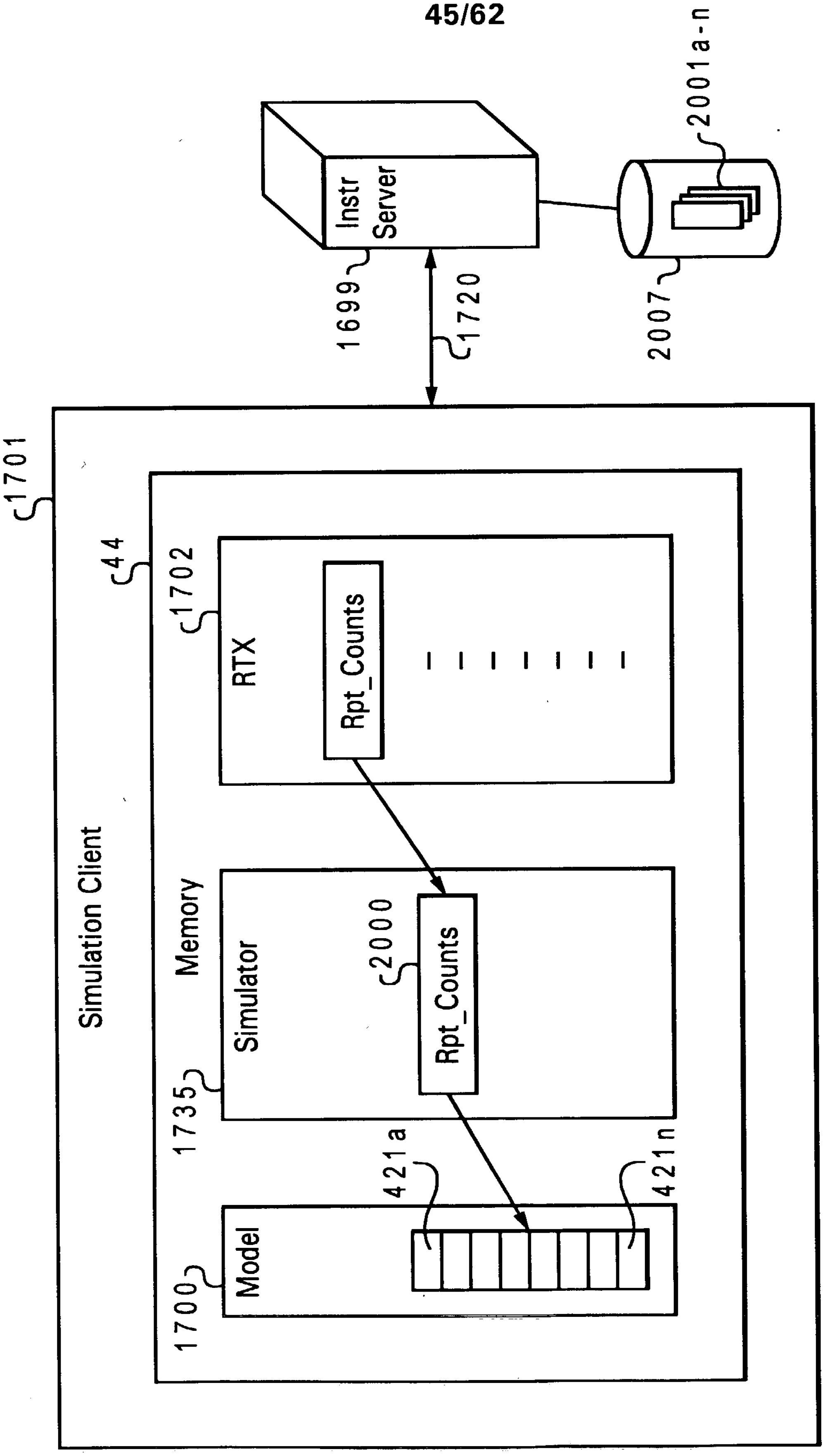

Instr Server
1699
1720
2001a-n
2007
Simulation Client
1701
44
Memory
1702
RTX
Rpt_Counts
Simulator
1735
2000
Rpt_Counts
Model
1700
421a
421n

Fig. 20A

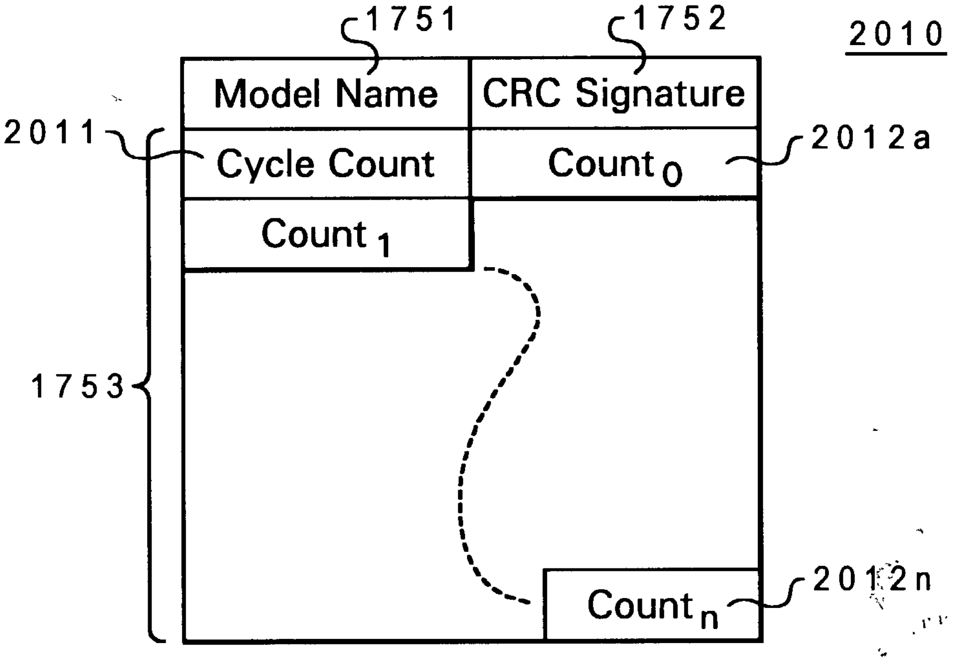


*Fig. 20B*

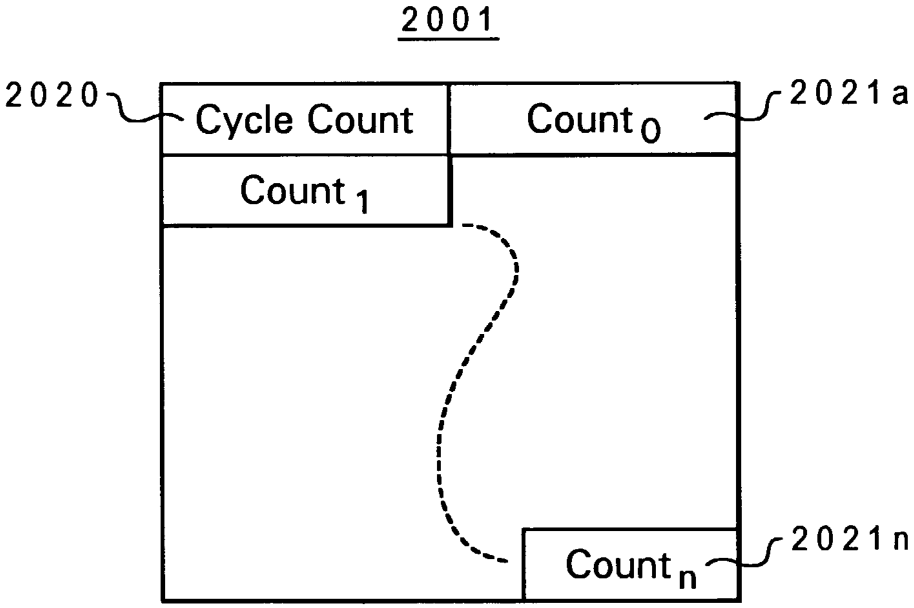


*Fig. 20C*

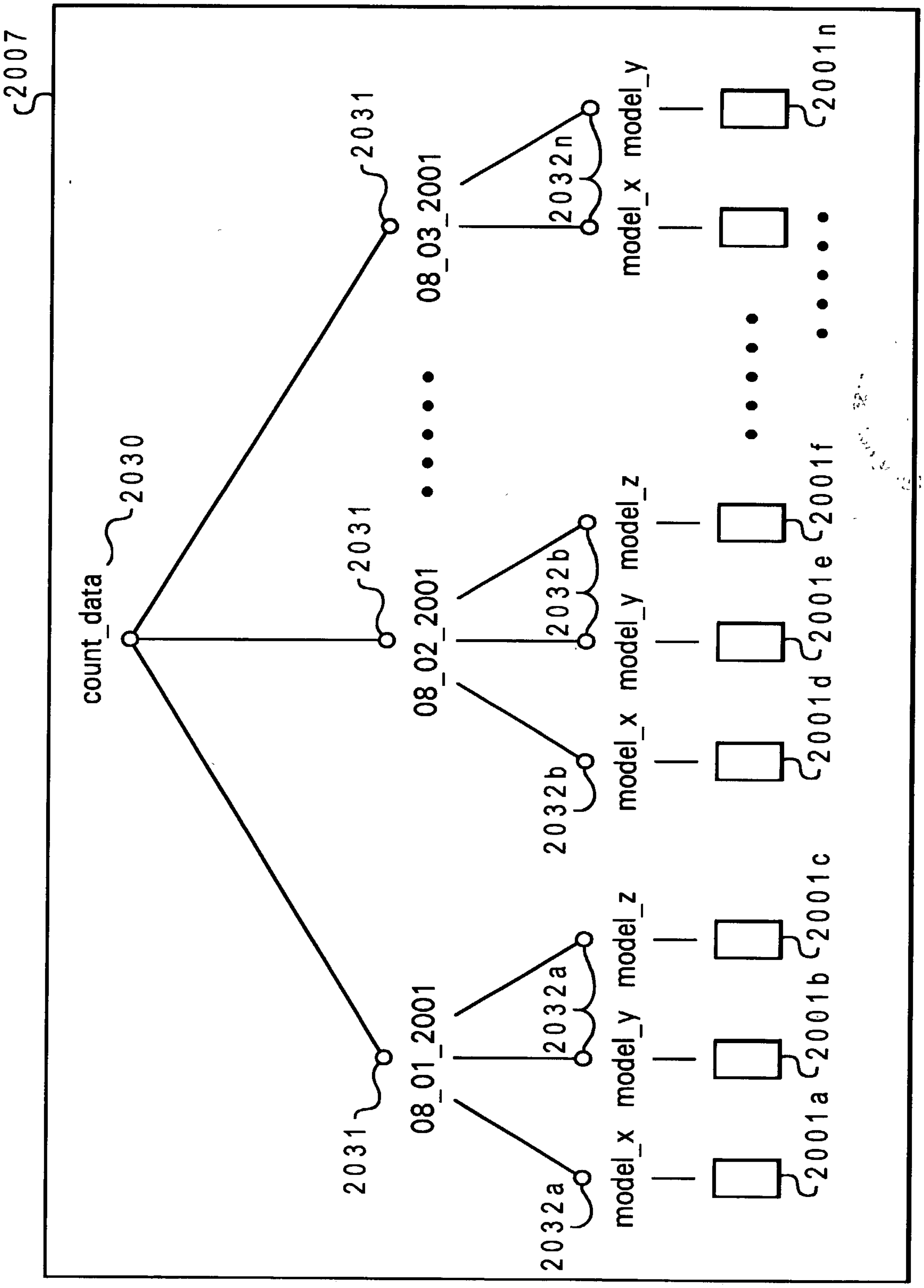

2007
count_data
2030
2031
08_01_2001
2031
08_02_2001
2031
08_03_2001
2032a
model_x
2032a
model_y
model_z
2032b
model_x
2032b
model_y
model_z
2032n
model_x
model_y
2001a
2001b
2001c
2001d
2001e
2001f
2001n

Fig. 20D

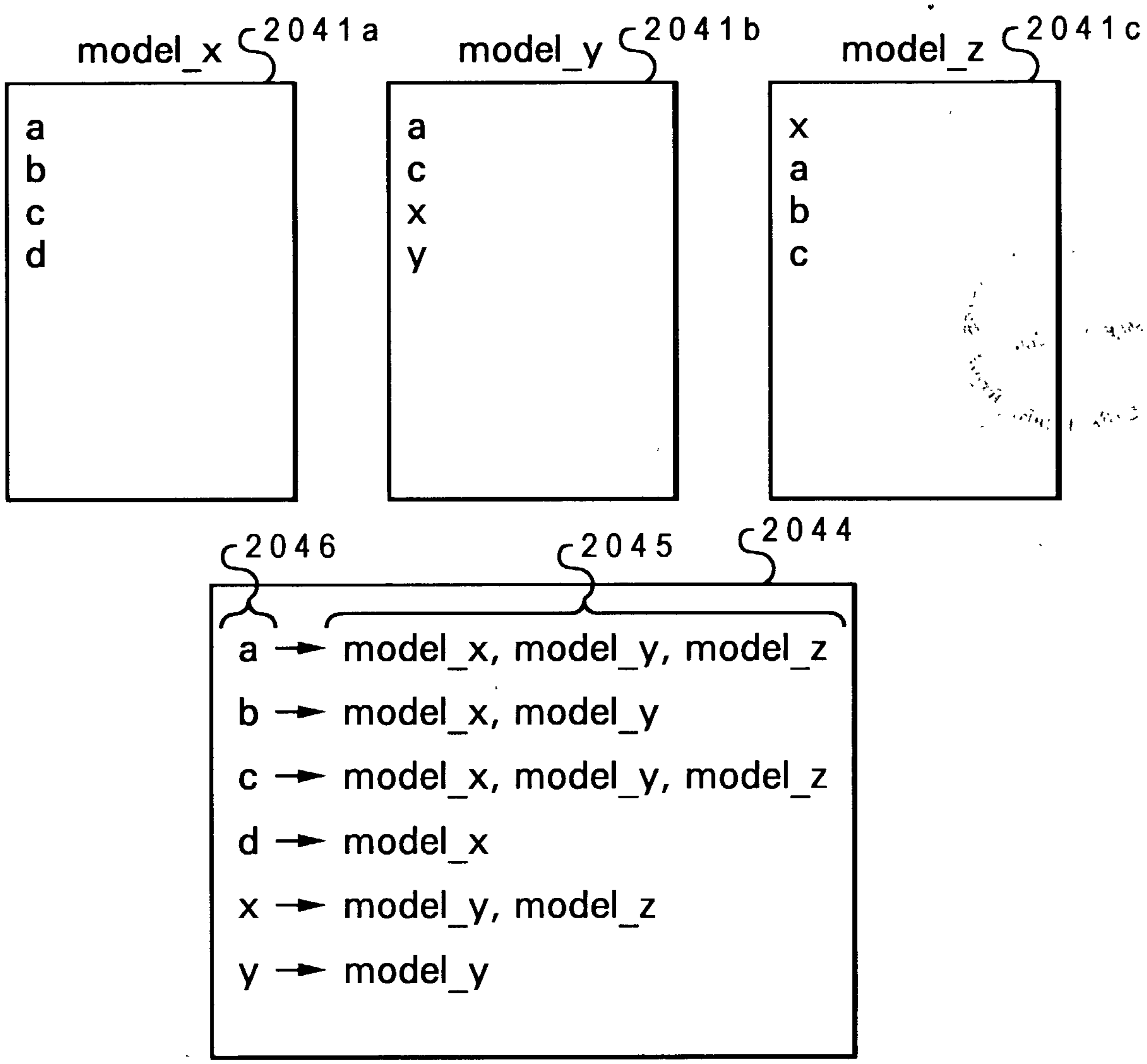

model_x
2041a
a
b
c
d
model_y
2041b
a
c
x
y
model_z
2041c
x
a
b
c
2046
2045
2044
a → model_x, model_y, model_z
b → model_x, model_y
c → model_x, model_y, model_z
d → model_x
x → model_y, model_z
y → model_y

*Fig. 20E*

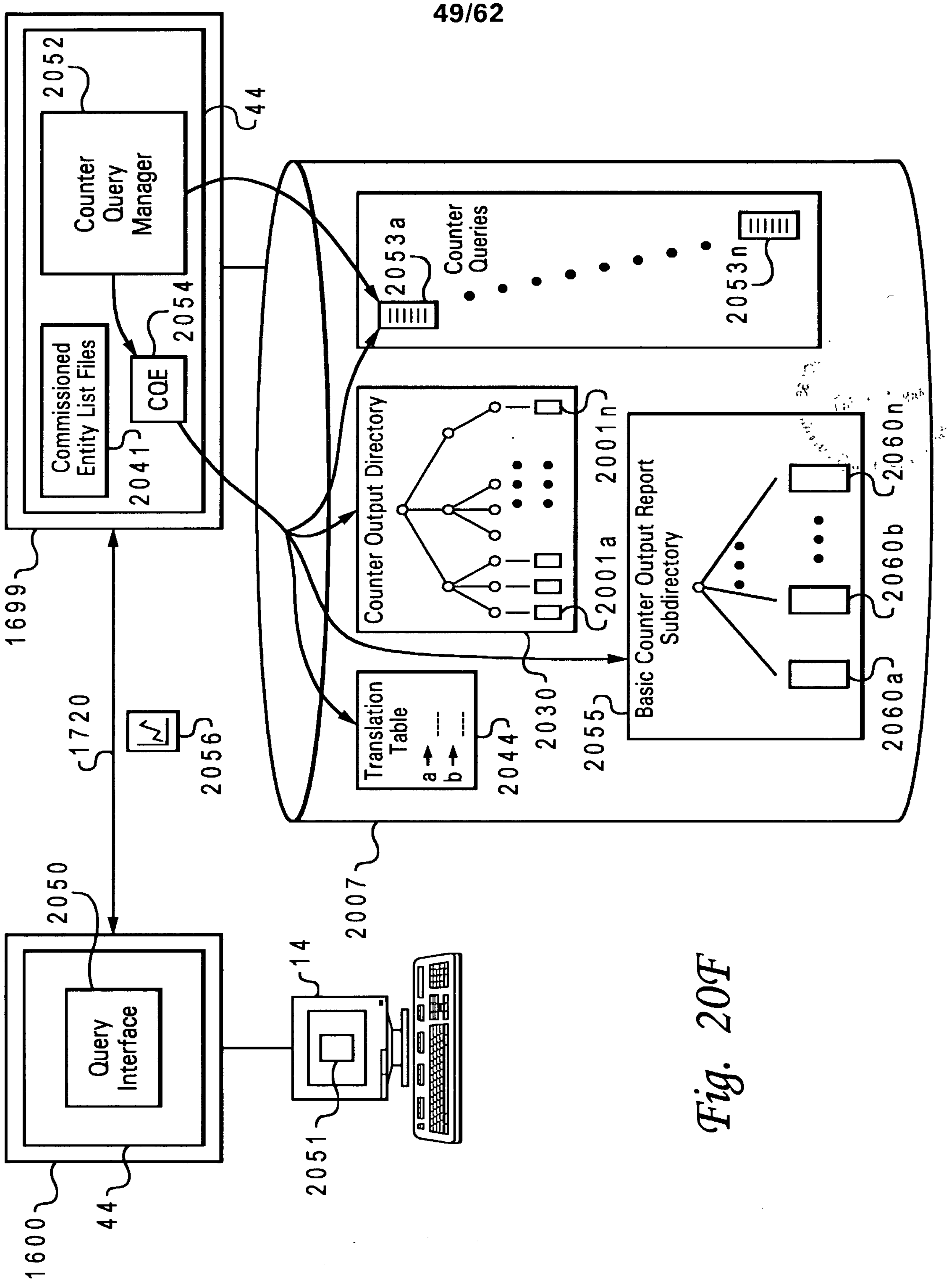
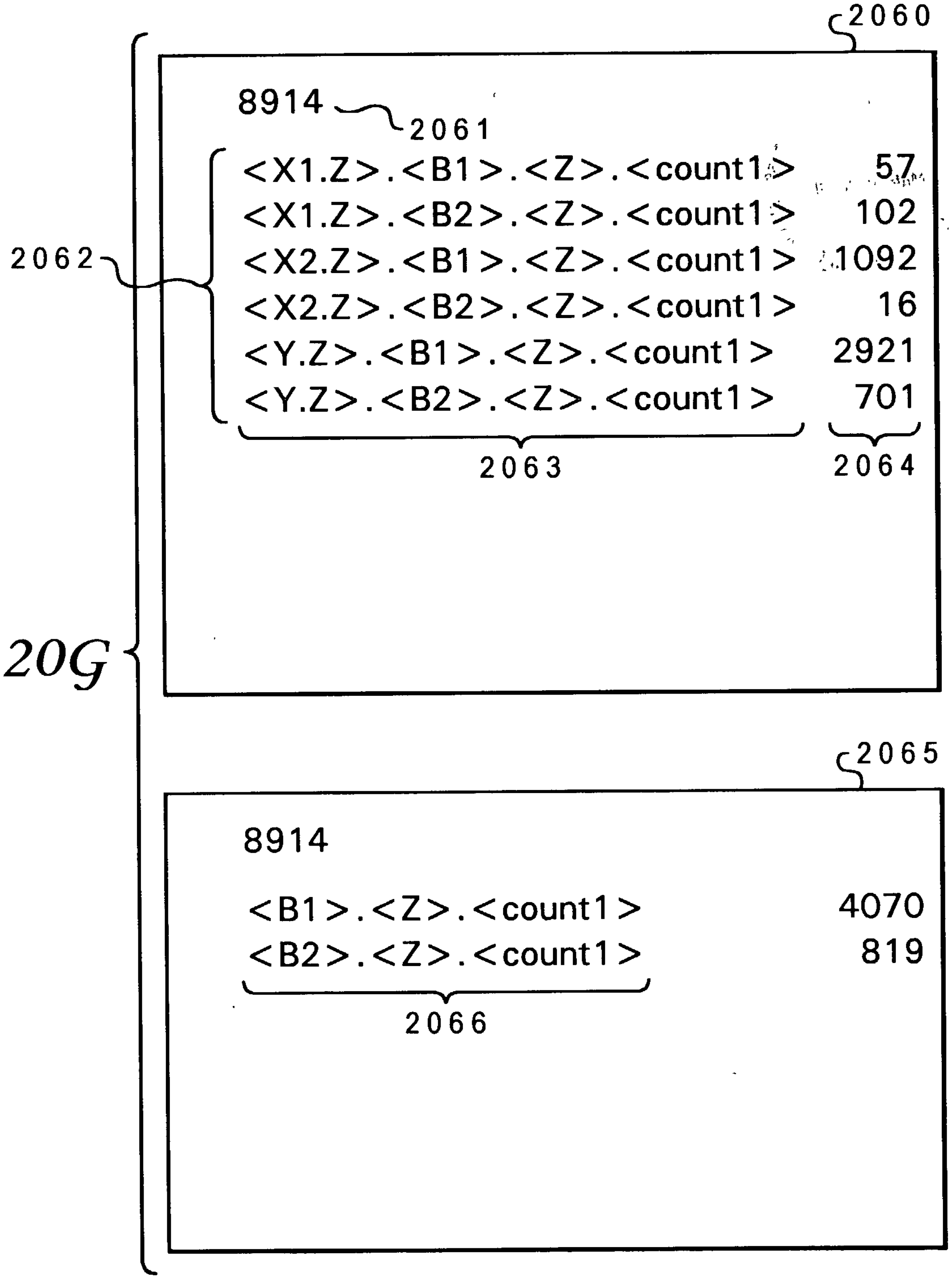

1699
44
2052
Counter Query Manager
Commissioned Entity List Files
2041
CQE
2054
1720
2056
2007
Counter Queries
2053a
2053n
Counter Output Directory
2001a
2001n
2030
Basic Counter Output Report Subdirectory
2055
2060a
2060b
2060n
Translation Table
a →
b →
2044
1600
44
2050
Query Interface
14
2051

Fig. 20F



8914

| | |
|---|---|
| <X1.Z>.<B1>.<Z>.<count1> | 57 |
| <X1.Z>.<B2>.<Z>.<count1> | 102 |
| <X2.Z>.<B1>.<Z>.<count1> | 1092 |
| <X2.Z>.<B2>.<Z>.<count1> | 16 |
| <Y.Z>.<B1>.<Z>.<count1> | 2921 |
| <Y.Z>.<B2>.<Z>.<count1> | 701 |

8914

| | |
|---|---|
| <B1>.<Z>.<count1> | 4070 |
| <B2>.<Z>.<count1> | 819 |

*Fig. 20G*

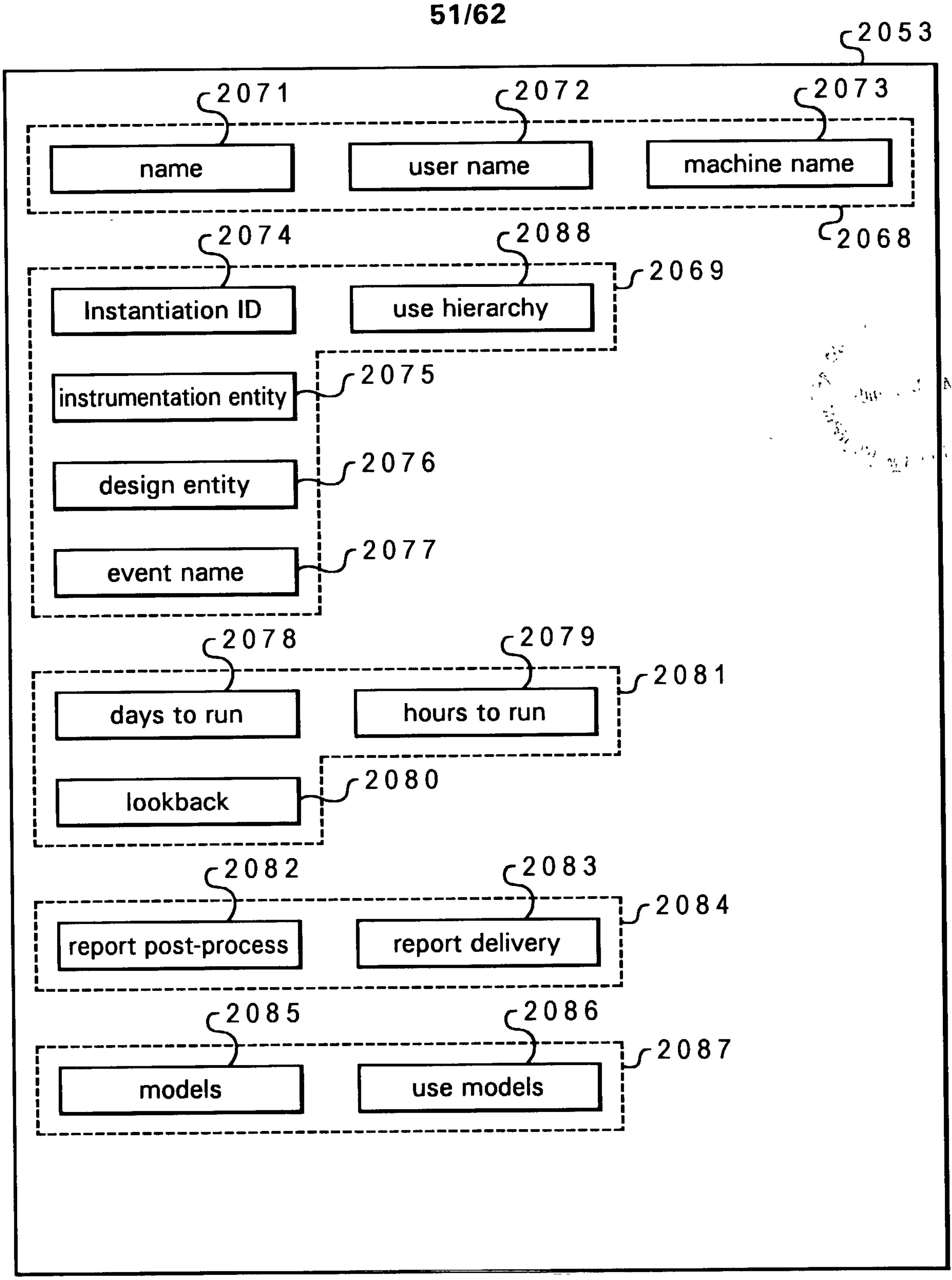

2053
2071
2072
2073
name
user name
machine name
2068
2074
2088
2069
Instantiation ID
use hierarchy
instrumentation entity
2075
design entity
2076
event name
2077
2078
2079
2081
days to run
hours to run
lookback
2080
2082
2083
2084
report post-process
report delivery
2085
2086
2087
models
use models

*Fig. 20H*

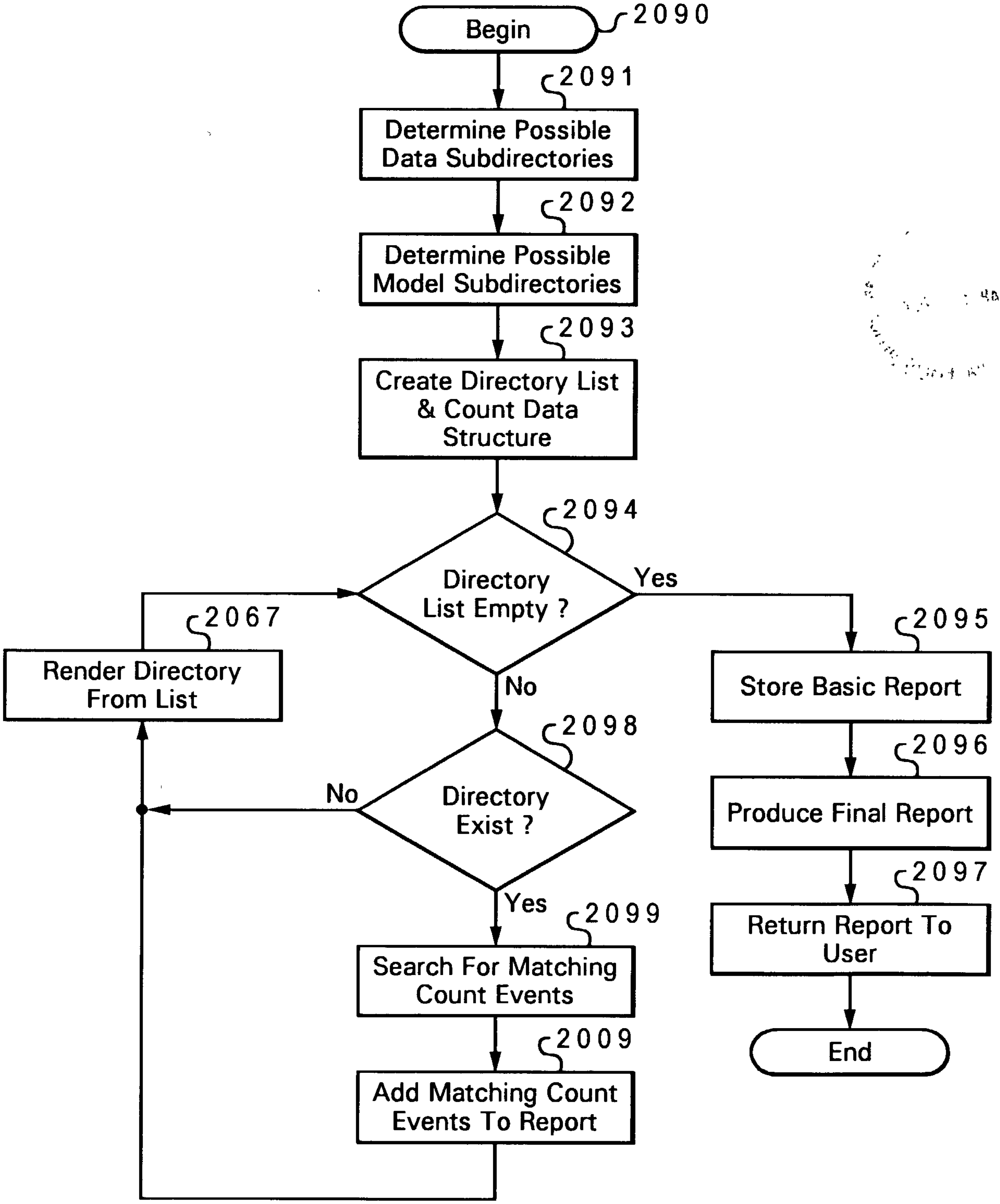

Begin
2090
2091
Determine Possible Data Subdirectories
2092
Determine Possible Model Subdirectories
2093
Create Directory List & Count Data Structure
2094
Directory List Empty ?
Yes
No
2067
Render Directory From List
2095
Store Basic Report
2096
Produce Final Report
2097
Return Report To User
End
2098
Directory Exist ?
No
Yes
2099
Search For Matching Count Events
2009
Add Matching Count Events To Report

*Fig. 20I*

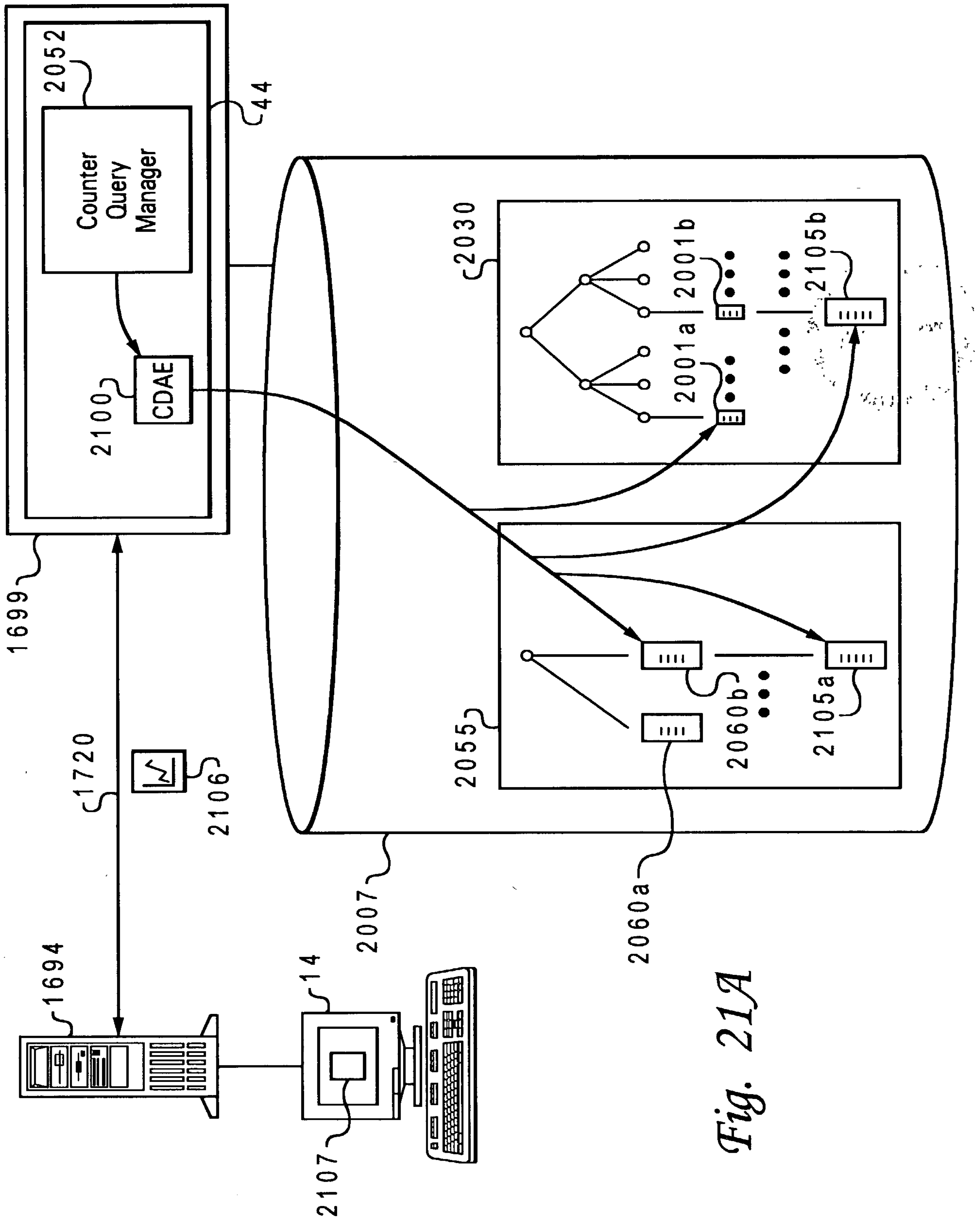

2052
Counter Query Manager
44
CDAE
2100
1699
2030
2001a
2001b
2105b
2055
2060b
2105a
2060a
2007
1720
2106
1694
14
2107

*Fig. 21A*

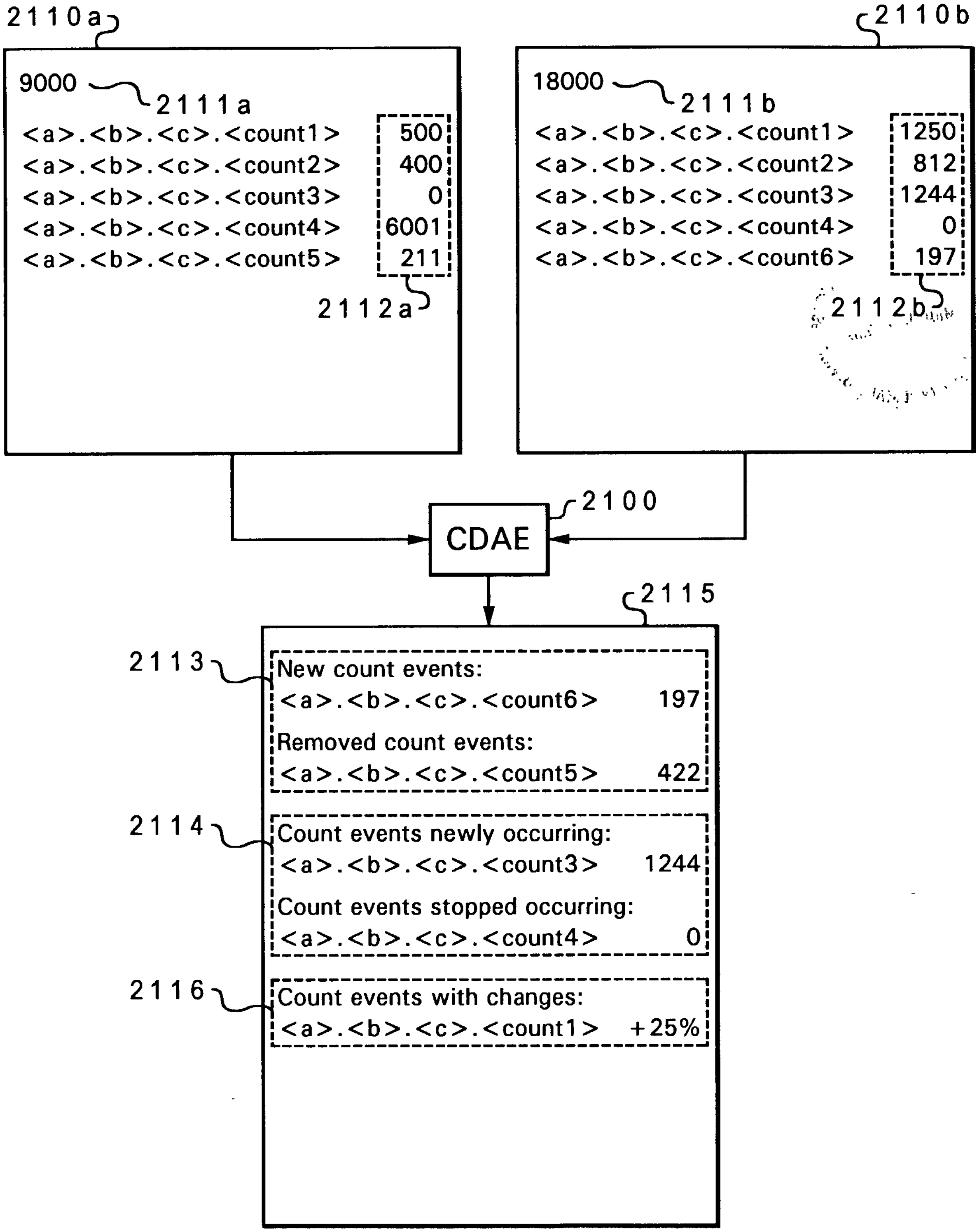

2110a
9000
2111a
<a>.<b>.<c>.<count1> 500
<a>.<b>.<c>.<count2> 400
<a>.<b>.<c>.<count3> 0
<a>.<b>.<c>.<count4> 6001
<a>.<b>.<c>.<count5> 211
2112a
2110b
18000
2111b
<a>.<b>.<c>.<count1> 1250
<a>.<b>.<c>.<count2> 812
<a>.<b>.<c>.<count3> 1244
<a>.<b>.<c>.<count4> 0
<a>.<b>.<c>.<count6> 197
2112b
CDAE
2100
2115
2113
New count events:
<a>.<b>.<c>.<count6> 197
Removed count events:
<a>.<b>.<c>.<count5> 422
2114
Count events newly occurring:
<a>.<b>.<c>.<count3> 1244
Count events stopped occurring:
<a>.<b>.<c>.<count4> 0
2116
Count events with changes:
<a>.<b>.<c>.<count1> +25%

*Fig. 21B*

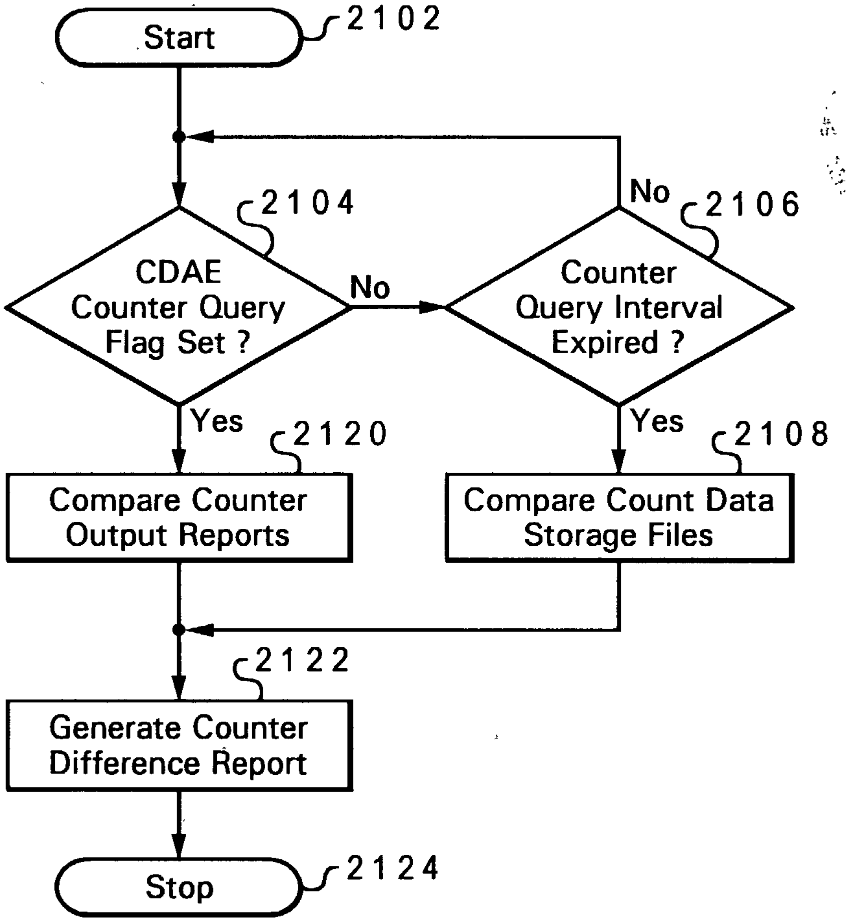

Start
2102
2104
CDAE Counter Query Flag Set ?
No
2106
Counter Query Interval Expired ?
No
Yes
2120
Compare Counter Output Reports
Yes
2108
Compare Count Data Storage Files
2122
Generate Counter Difference Report
Stop
2124

*Fig. 21C*

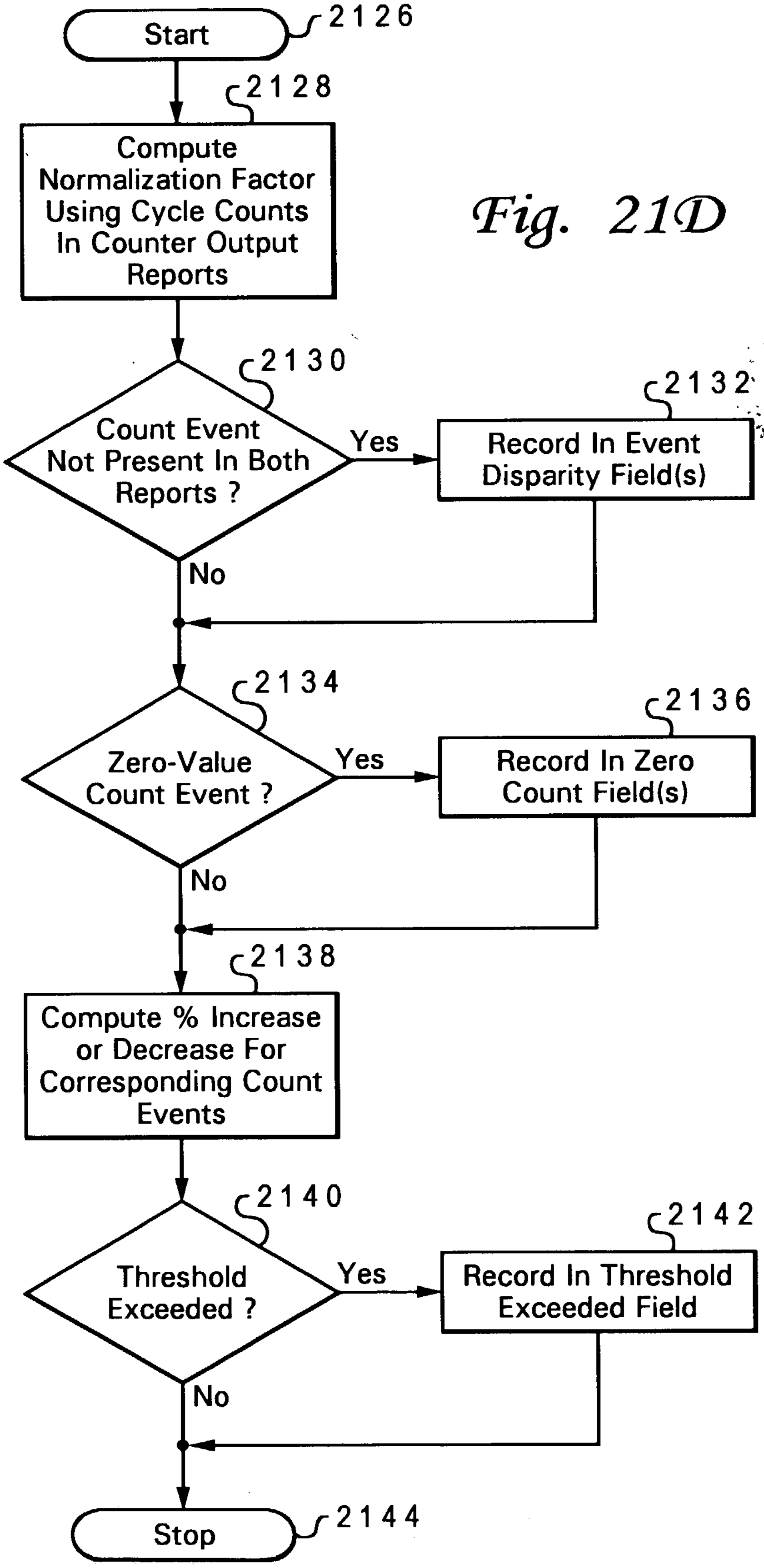

Start
2126
2128
Compute Normalization Factor Using Cycle Counts In Counter Output Reports
2130
Count Event Not Present In Both Reports ?
Yes
2132
Record In Event Disparity Field(s)
No
2134
Zero-Value Count Event ?
Yes
2136
Record In Zero Count Field(s)
No
2138
Compute % Increase or Decrease For Corresponding Count Events
2140
Threshold Exceeded ?
Yes
2142
Record In Threshold Exceeded Field
No
Stop
2144

*Fig. 21D*

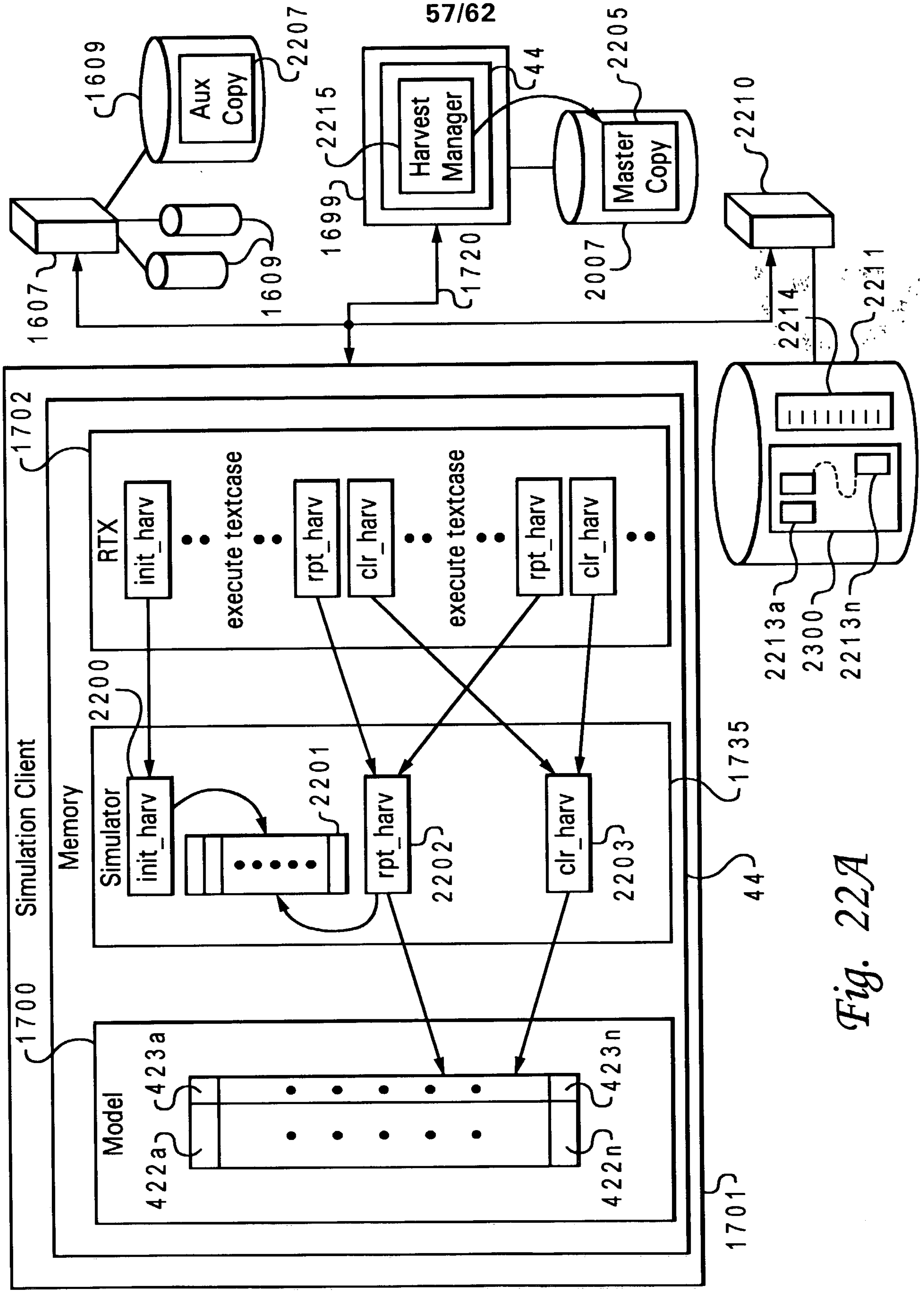

1609
Aux Copy
2207
2215
Harvest Manager
44
2205
Master Copy
2210
1607
1609
1699
1720
2007
2214
2211
Simulation Client
1702
RTX
init_harv
execute textcase
rpt_harv
clr_harv
execute textcase
rpt_harv
clr_harv
2213a
2300
2213n
Memory
Simulator
2200
init_harv
2201
rpt_harv
2202
clr_harv
2203
1735
44
1700
Model
423a
422a
423n
422n
1701

Fig. 22A

Fig. 22B

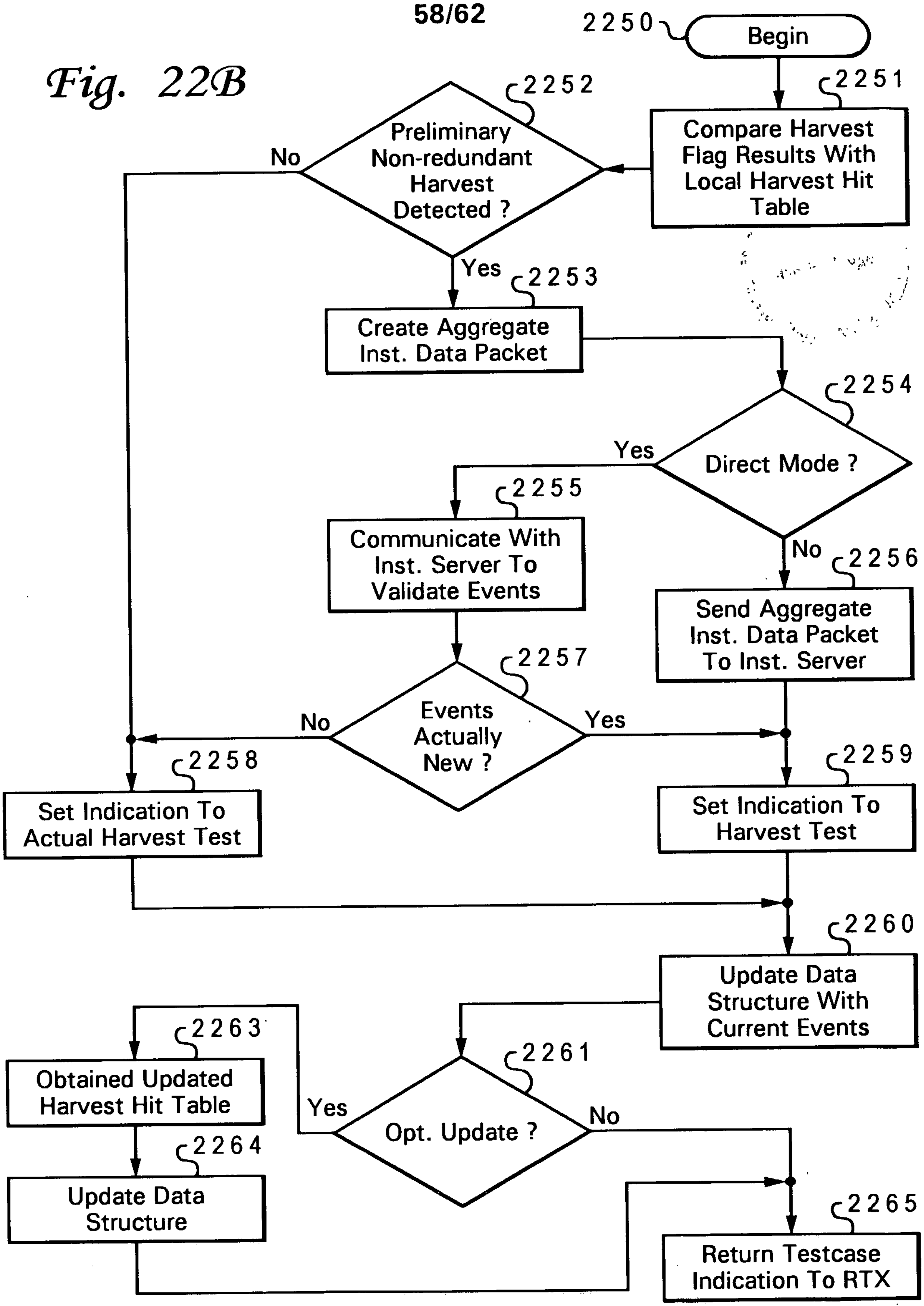
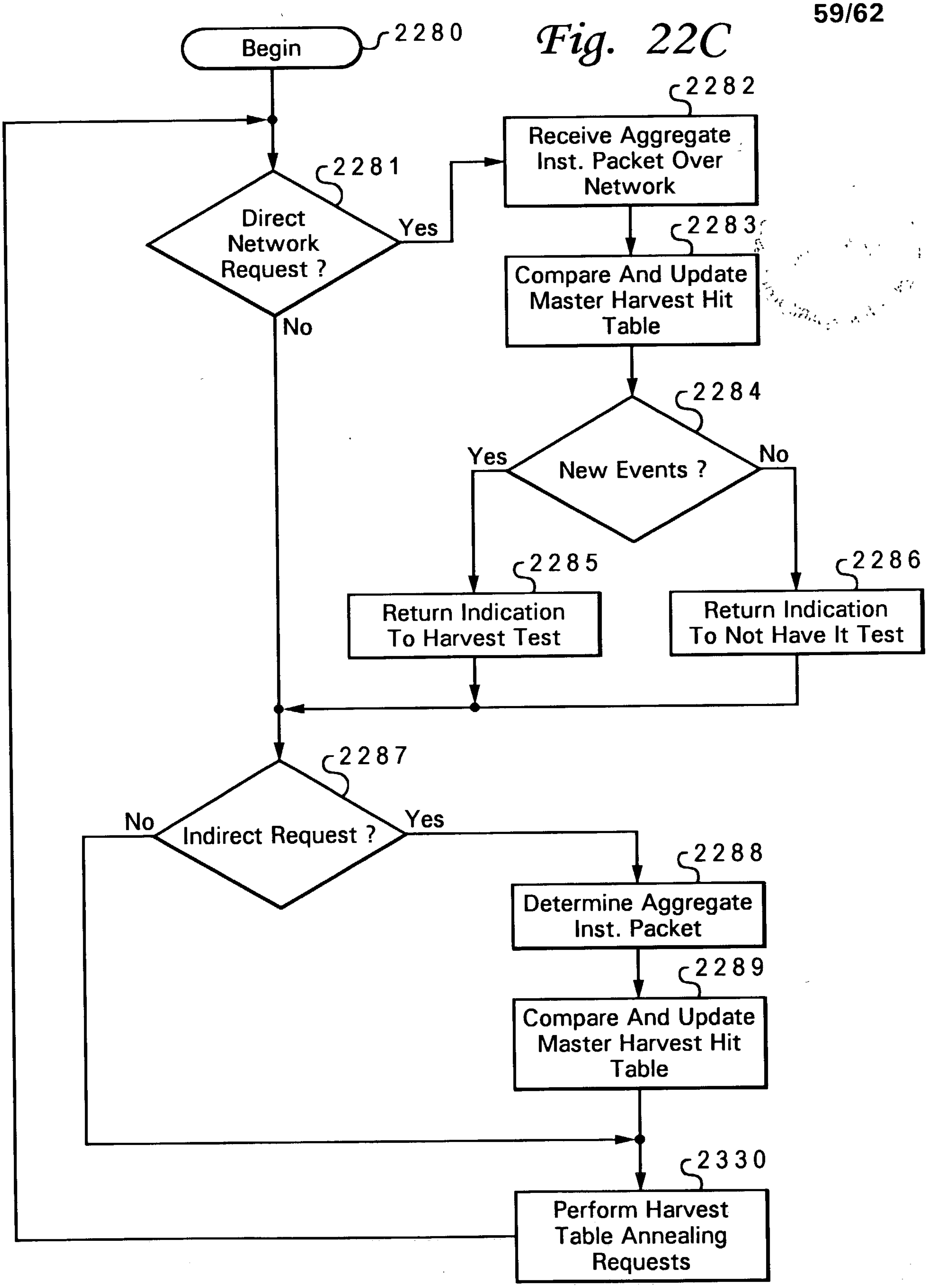

2250
Begin
2251
Compare Harvest Flag Results With Local Harvest Hit Table
2252
Preliminary Non-redundant Harvest Detected ?
No
Yes
2253
Create Aggregate Inst. Data Packet
2254
Direct Mode ?
Yes
No
2255
Communicate With Inst. Server To Validate Events
2256
Send Aggregate Inst. Data Packet To Inst. Server
2257
Events Actually New ?
No
Yes
2258
Set Indication To Actual Harvest Test
2259
Set Indication To Harvest Test
2260
Update Data Structure With Current Events
2261
Opt. Update ?
Yes
No
2263
Obtained Updated Harvest Hit Table
2264
Update Data Structure
2265
Return Testcase Indication To RTX

*Fig. 22C*


Begin
2280
2281
Direct Network Request ?
Yes
No
2282
Receive Aggregate Inst. Packet Over Network
2283
Compare And Update Master Harvest Hit Table
2284
New Events ?
Yes
No
2285
Return Indication To Harvest Test
2286
Return Indication To Not Have It Test
2287
Indirect Request ?
No
Yes
2288
Determine Aggregate Inst. Packet
2289
Compare And Update Master Harvest Hit Table
2330
Perform Harvest Table Annealing Requests

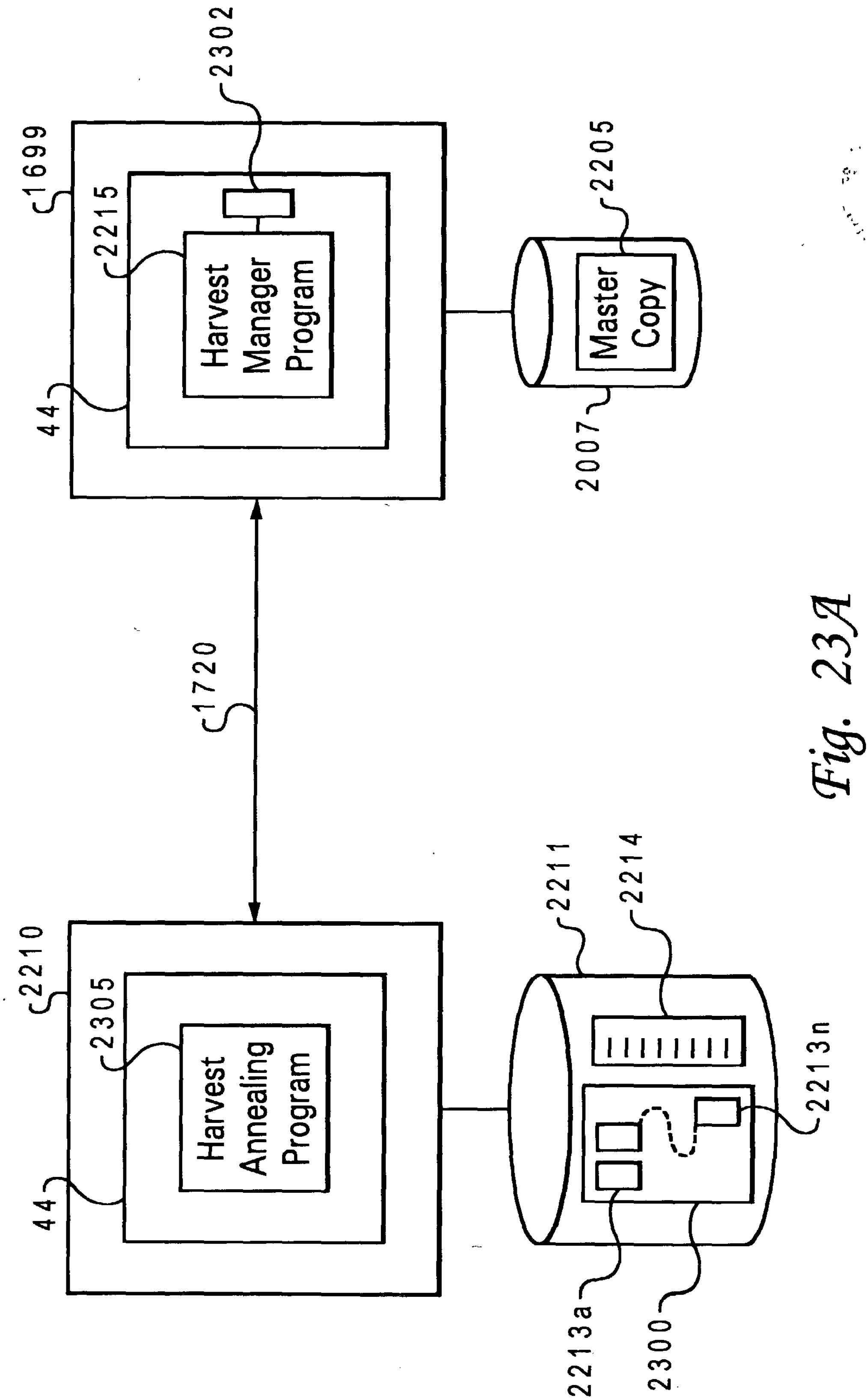
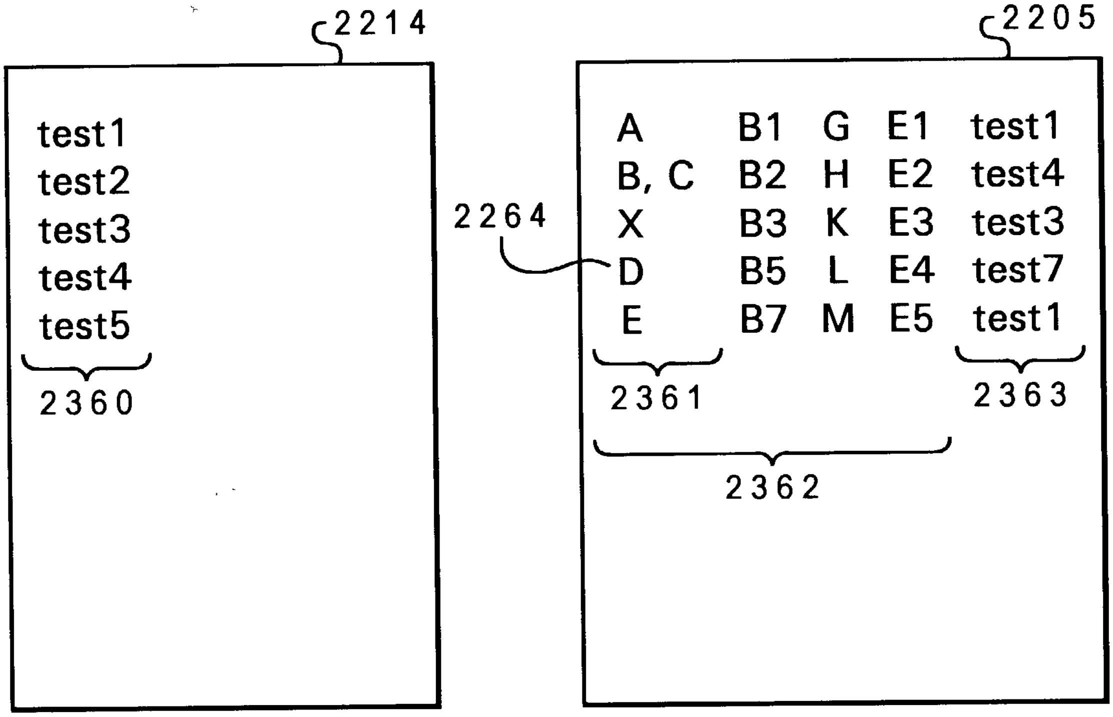

1699
2302
2215
44
Harvest Manager Program
2205
Master Copy
2007
1720
2210
2305
44
Harvest Annealing Program
2211
2214
2213n
2213a
2300

*Fig. 23A*



test1
test2
test3
test4
test5

| | | | | |
|---|---|---|---|---|
| A | B1 | G | E1 | test1 |
| B, C | B2 | H | E2 | test4 |
| X | B3 | K | E3 | test3 |
| D | B5 | L | E4 | test7 |
| E | B7 | M | E5 | test1 |

*Fig. 23B*

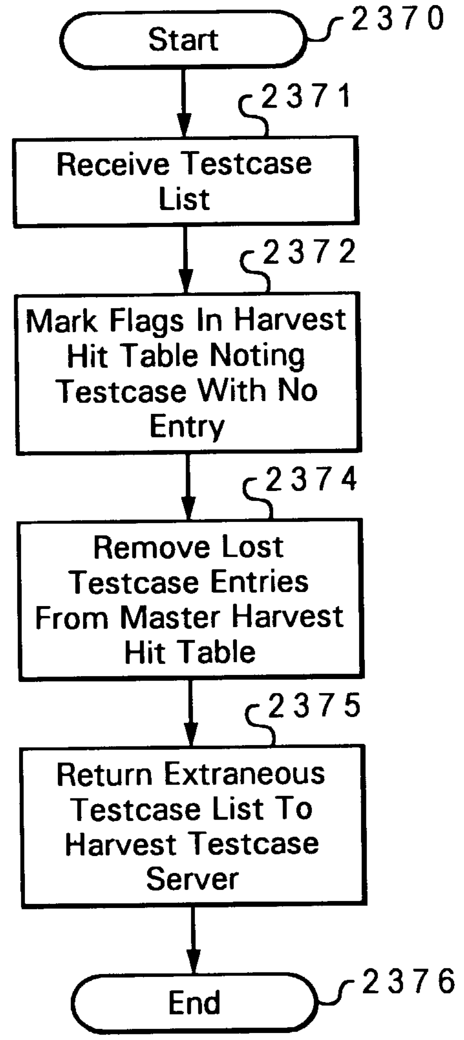

Start
2370
2371
Receive Testcase List
2372
Mark Flags In Harvest Hit Table Noting Testcase With No Entry
2374
Remove Lost Testcase Entries From Master Harvest Hit Table
2375
Return Extraneous Testcase List To Harvest Testcase Server
End
2376

*Fig. 23C*